

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

DEPARTMENT OF ELECTRICAL ENGINEERING  
SCHOOL OF ENGINEERING  
OLD DOMINION UNIVERSITY  
NORFOLK, VIRGINIA

DESIGN OF MULTIVARIABLE FEEDBACK CONTROL  
SYSTEMS VIA SPECTRAL ASSIGNMENT

By

Roland R. Mielke, Principal Investigator

Leonard J. Tung, Co-Principal Investigator

and

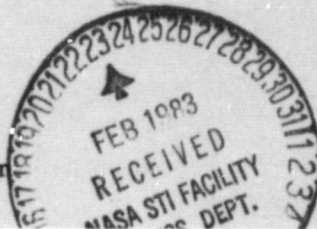
Mohsen Marefat

Progress Report

For the period March 1, 1982 to September 30, 1982

Prepared for the  
National Aeronautics and Space Administration  
Langley Research Center  
Hampton, Virginia 23665

Under  
NASA Grant NSG-1650  
Ruben Jones, Technical Monitor  
Flight Dynamics and Control Division



(NASA-CR-169889) DESIGN OF MULTIVARIABLE FEEDBACK CONTROL SYSTEMS VIA SPECTRAL ASSIGNMENT Progress Report, 1 Mar. - 30 Sep. 1982 (Old Dominion Univ., Norfolk, Va.) 112 p HC A06/MF A01 N83-1833 Unclassified CSCL 09B G3/63 02789



DEPARTMENT OF ELECTRICAL ENGINEERING  
SCHOOL OF ENGINEERING  
OLD DOMINION UNIVERSITY  
NORFOLK, VIRGINIA

DESIGN OF MULTIVARIABLE FEEDBACK CONTROL  
SYSTEMS VIA SPECTRAL ASSIGNMENT

By

Roland R. Mielke, Principal Investigator

Leonard J. Tung, Co-Principal Investigator

and

Mohsen Marefat

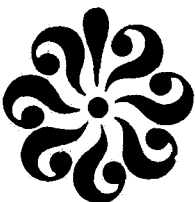
Progress Report

For the period March 1, 1982 to September 30, 1982

Prepared for the  
National Aeronautics and Space Administration  
Langley Research Center  
Hampton, Virginia 23665

Under  
NASA Grant NSG-1650  
Ruben Jones, Technical Monitor  
Flight Dynamics and Control Division

Submitted by the  
Old Dominion University Research Foundation  
P.O. Box 6369  
Norfolk, Virginia 23508-0369



January 1983

## ABSTRACT

This report summarizes the progress of applied research conducted under NASA Grant NSG-1650 during the period March 1, 1982 to September 30, 1982. The objective of this project is to investigate the applicability of spectral assignment techniques to the design of multivariable feedback control systems. A fractional representation design procedure for unstable plants is presented and illustrated with an example. Then, a computer aided design software package implementing eigenvalue/eigenvector design procedures is described. A design example which illustrates the use of the program is explained.

**PRECEDING PAGE BLANK NOT FILMED**

## TABLE OF CONTENTS

	<u>Page</u>
<b>ABSTRACT</b>	
1. INTRODUCTION . . . . .	1
2. FRACTIONAL REPRESENTATION DESIGN PROCEDURES . . . . .	1
2.1. Introduction . . . . .	1
2.2. Compensator Design . . . . .	2
2.3. Complexity of Controllers . . . . .	6
2.4. Design Example . . . . .	7
2.5. Hidden Modes . . . . .	9
3. EIGENVALUE/EIGENVECTOR ASSIGNMENT PROCEDURES . . . . .	11
3.1. Introduction . . . . .	11
3.2. Design Philosophy . . . . .	12
3.3. Computer Aided Design Software Package . . . . .	14
3.4. Design Example . . . . .	17
4. REFERENCES . . . . .	37
APPENDIX . . . . .	39

# DESIGN OF MULTIVARIABLE FEEDBACK CONTROL SYSTEMS VIA SPECTRAL ASSIGNMENT

By

Roland R. Mielke<sup>1</sup>, Leonard J. Tung<sup>2</sup> and Mohsen Marefat<sup>3</sup>

## 1. INTRODUCTION

This report summarizes the progress of applied research conducted under NASA Grant NSG-1650 for the period March 1, 1982 to September 30, 1982. The objective of this work is to investigate the applicability of spectral assignment techniques to the design of multivariable feedback control systems.

First, development of new frequency domain fractional representation design procedures for unstable plants is presented. The procedure consists of a technique for searching among all stabilizing controllers for those that also satisfy certain design specifications. Controller complexity and hidden system modes are considered. The procedure is illustrated with a design example. Then a new computer aided design software package implementing the time domain eigenvalue/eigenvector assignment procedures is described. The use of the program is illustrated with a design example. The program listing is included in the Appendix.

## 2. FRACTIONAL REPRESENTATION DESIGN PROCEDURES

### 2.1. Introduction

Our investigation in the area of frequency-domain controller design began with a study of the work by Youla and others (refs. 1,2). Among the many contributions in Youla's work is a procedure which leads to the

---

<sup>1</sup> Associate Professor, Department of Electrical Engineering, Old Dominion University, Norfolk, Virginia 23508.

<sup>2</sup> Assistant Professor, Department of Electrical Engineering, Old Dominion University, Norfolk, Virginia 23508.

<sup>3</sup> Graduate Research Assistant, Department of Electrical Engineering, Old Dominion University, Norfolk, Virginia 23508.

characterization of a general class of stabilizing compensators for a plant imbedded in a single-loop feedback control system. This procedure has then been generalized by Desoer and others (ref. 3) to form the basis of the so-called fractional representation approach. This approach offers a systematic procedure for constructing stabilizing compensators that achieve other design objectives such as decoupling the outputs and tracking step inputs. It should be noted that the objective of stabilization is resolved before other design objectives. In contrast to this type of approach is the work by Sain and others (refs. 4,5). Sain's work develops a direct method for the construction of compensators for a plant imbedded in a unity feedback control system. In this method, compensators that achieve design objectives such as decoupling are first constructed and then the issue of stabilization is resolved. Combining the results by Desoer and by Sain, we have developed design procedures that simultaneously achieve the design objectives of stabilization, decoupling, and tracking step inputs. These design procedures are expressed so that it is relatively easy to address the problems of complex compensators and unwanted hidden modes as noted in references 5-7.

In this report, we begin with a brief review of the fractional representation approach. After the review we outline two sets of procedures, one for stable plants and one for unstable plants, for constructing compensators that achieve design objectives of stabilization, decoupling, and tracking step inputs. These design procedures also allow us to construct simple compensators  $C = -P^{-1} T(I-T)^{-1}$  for a given plant  $P$  by choosing simple stable diagonal  $T$  which satisfies certain requirements. The details of the procedures are exemplified by a problem of compensator design for an unstable plant. Finally, the problem of hidden modes is dealt with by carefully choosing the zeros of  $I-T$ .

## 2.2. Compensator Design

Consider the single feedback loop multivariable control system shown in figure 1. With the plant  $P(s)$  (a proper rational matrix) given, it is desired to design a controller  $C(s)$  (another proper rational matrix) for stabilization, decoupling and tracking step inputs. The fractional representation approach (refs. 3,8,9) offers a systematic procedure for

achieving these design objectives. In this approach, the plant is expressed in a right and a left coprime exponential stable rational fractional descriptions,  $P = N_r D_r^{-1} = D_l^{-1} N_l$  with  $U_r, V_r, U_l$  and  $V_l$  such that  $U_r N_r + V_r D_r = N_l U_l + D_l V_l = I$ . Note that all terms except possibly the plant  $P$  are proper rational matrices with poles in the open left-half complex plane. With these descriptions for  $P$ , a general class of stabilizing controllers is given by

$$C = (W N_l + V_r)^{-1} (-W D_l + U_r), \quad (1)$$

where  $W$  can be any proper exponential stable rational matrix as long as  $W N_l + V_r$  is nonsingular. With this class of controllers, the closed-loop transfer function is given by

$$T = N_r [-W D_l + U_r]. \quad (2)$$

When  $P$  itself is exponential stable, we can have  $N_r = N_l = P$ ,  $D_r = D_l = I$ ,  $U_r = U_l = 0$  and  $V_r = V_l = I$ . Equations (1) and (2) then become

$$C = -W[I+PW]^{-1} \quad (3)$$

and

$$T = -PW. \quad (4)$$

Equations (1) through (4) display the freedom in choosing a stabilizing controller as the freedom in choosing  $W$ . This freedom in choosing  $W$  can then be explored for achieving other design objectives such as decoupling and tracking. For decoupling,  $T$  is to be made diagonal. For tracking step inputs, every term in  $I-T$  should have a zero at  $s = 0$ .

ORIGINAL PAGE IS  
OF POOR QUALITY



ORIGINAL PAGE IS  
OF POOR QUALITY

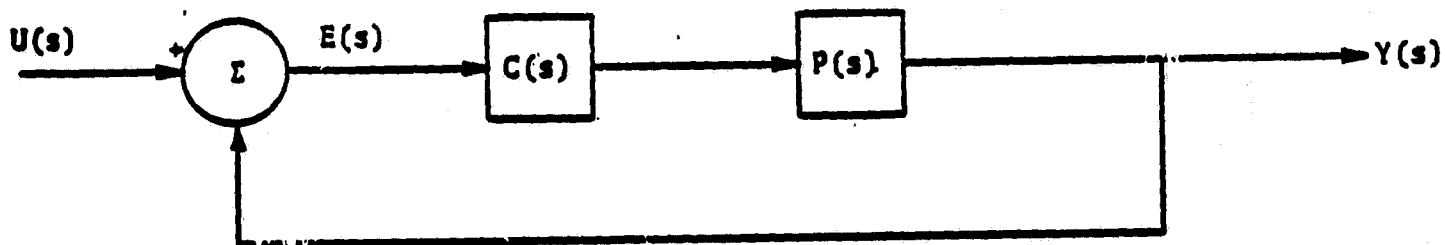


Figure 1. Multivariable feedback control systems.

### Stable Plants

First assume that  $P$  is exponential stable and invertible. Under this assumption,  $T = -PW$  is invertible if  $W$  is invertible. The invertibility of  $T$  is important because it eliminates the possibility of zero diagonal terms in  $T$  after  $T$  is made diagonal for decoupling, a case which indicates redundancy of certain input and output signals. For maintaining the stability of the closed-loop system, our approach is to choose stable  $T$  to make  $W = -P^{-1}T$  stable. For decoupling, we only have to work with diagonal  $T$ . For tracking step inputs, we must choose among those matrices  $T$  such that all terms in  $I-T$  have a zero at  $s = 0$ . For constructing controllers that simultaneously achieve stabilization, decoupling and tracking step inputs, we thus have the following procedures:

(i) For decoupling, choose  $T = \text{diag} \{T_1, \dots, T_n\}$ .

(ii) Let  $P^{-1} = [\tilde{P}_1, \dots, \tilde{P}_n]$ , where  $\tilde{P}_i$  is the  $i$ th column of  $P^{-1}$ . We then have  $P^{-1}T = [\tilde{P}_1 T_1, \dots, \tilde{P}_n T_n]$ . For maintaining the stability, each  $T_i$  and each  $\tilde{P}_i T_i$  should be proper. Poles of  $T_i$  should be in the open left-half complex plane. Zeros of  $T_i$  must cancel the closed right-half plane poles of  $\tilde{P}_i$ .

(iii) Let  $T_i = n_i/d_i$ . For tracking step inputs, each  $(d_i - n_i)$  should have a zero at  $s = 0$ , i.e. no constant term.

(iv)  $C = -P^{-1}T(1-T)^{-1}$ .

For unstable plants, similar design procedures can be derived. Again, we are interested in invertible matrices  $T$ . This requires the assumption that  $P$  is invertible, which in turn implies that  $N_r$  and  $N_\ell$  are invertible. As before, we use diagonal  $T$  for decoupling and we choose those matrices  $T$  such that all terms in  $I-T$  have a zero at  $s = 0$  for tracking step inputs. For stabilization, however, we choose stable  $T$  to make  $W = (-N_r^{-1}T + U_r)D_\ell^{-1}$  stable. This process is more involved than the corresponding process for stable plants. The reason for this is that  $U_r D_\ell^{-1}$  may be unstable for a given unstable plant. In order to achieve stability, somehow part of  $N_r^{-1} T D_\ell^{-1}$  is to be made unstable to cancel the unstable part of  $U_r D_\ell^{-1}$ . With this in mind, we have the following design procedures for constructing controllers that simultaneously achieve stabilization, decoupling and tracking of step inputs:

(i) For decoupling, choose  $T = \text{diag} \{T_1, \dots, T_n\}$

(ii) Find a stable  $T_0 = \text{diag} \{T_{01}, \dots, T_{0n}\}$

to make  $-N_r^{-1} T_0 D_\ell^{-1} + U_r D_\ell^{-1}$  stable

(iii) Let  $T_s = \text{diag} \{T_{s1}, \dots, T_{sn}\}$ ,  $N_r^{-1} = [\tilde{N}_1, \dots, \tilde{N}_n]$

and  $D_\ell^{-1} = \begin{bmatrix} \tilde{D}_1 \\ \vdots \\ \tilde{D}_n \end{bmatrix}$ , where  $\tilde{N}_i$  is the  $i$ th column of  $N_r^{-1}$  and  $\tilde{D}_j$  is

the  $j$ th row of  $D_\ell^{-1}$ . We then have

$$N_r^{-1} T_s D_\ell^{-1} = \sum_{i=1}^n \tilde{N}_i T_{si} \tilde{D}_i$$

ORIGINAL PAGE IS  
OF POOR QUALITY

For achieving stability, each  $T_{si}$  and  $\tilde{N}_i T_{si} \tilde{D}_i$  should be proper. Poles of  $T_{si}$  should be in the open left-half complex plane. Zeros of  $T_{si}$  must cancel the closed right-half plane poles of  $\tilde{N}_i$  and  $\tilde{D}_i$ .

(iv) Let  $T_{oi} = n_{oi}/d_{oi}$  and  $T_{si} = n_{si}/d_{si}$ .

For tracking step inputs, each  $(d_{si}d_{oi} - n_{si}d_{oi} - n_{oi}d_{si})$  should have a zero at  $s = 0$ , i.e. no constant term.

(v)  $T = T_o + T_s$ , and

$$C = -PT(I-T)^{-1}.$$

### 2.3. Complexity of Controllers

As pointed out earlier, the fractional representation approach allows us to search systematically for compensators that achieve various design objectives. This approach, however, does not always bring about simple compensators. As a matter of fact, the time-domain dynamic compensation (ref. 10) is more likely to bring about simple stabilizing compensators than the fractional representation approach. On the other hand, the fractional representation approach is more likely to result in simple stabilizing compensators that also decouple system outputs. The latter is due to the difficulty in dynamic compensation of relating directly the diagonality of a transfer fraction to the formation of the system matrices  $\{A, B, C, D\}$  in the state-space description of a system.

In our design procedures, the compensator  $C$  is given by  $C = -P^{-1}T(I-T)^{-1}$ . For a given plant  $P$ ,  $T$  is to be chosen for forming compensators that stabilize the system, decouple the outputs as well as track step inputs. Under close examination, we notice that the poles of  $T$  will basically be cancelled by the same poles of  $(I - T)$  in forming  $C$ . These poles do not directly affect the complexity of the controllers. However, the total number of the poles determines the degree of freedom in choosing the

zeros of  $T$  and  $I-T$ . For simple compensators, the zeros of  $T$  can be chosen to cancel the poles of  $P^{-1}$ , and zeros of  $I-T$  can be chosen to cancel the zeros of  $P^{-1}$ . Overall,  $T$  should be kept as simple as possible. The following problem illustrates the details involved. This problem was first discussed in references 6 and 7.

#### 2.4. Design Example

For a plant

$$P(s) = \begin{bmatrix} \frac{1}{s+1} & \frac{s-1}{s+1} \\ 0 & \frac{1}{s-1} \end{bmatrix}$$

We have derived a set of stable matrices  $N_r, D_r, N_\ell, D_\ell, U_r, V_r, U_\ell$  and  $V_\ell$  given by

$$D_r = D_\ell = \begin{bmatrix} 1 & 0 \\ 0 & \frac{s-1}{s+2} \end{bmatrix}, \quad N_r = \begin{bmatrix} \frac{1}{s+1} & \frac{(s-1)^2}{(s+1)(s+2)} \\ 0 & \frac{1}{s+2} \end{bmatrix}$$

$$N_\ell = \begin{bmatrix} \frac{1}{s+1} & \frac{s-1}{s+1} \\ 0 & \frac{1}{s+2} \end{bmatrix}, \quad V_r = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$U_r = U_\ell = \begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}, \quad V_\ell = \begin{bmatrix} 1 & \frac{-3(s-1)}{s+1} \\ 0 & 1 \end{bmatrix}$$

Since

$$U_r D_\ell^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{3(s+2)}{s-1} \end{bmatrix}$$

is not stable, a simple stable  $T_0$  is to be found to make

$$-N_r^{-1} T_0 D_\ell^{-1} + U_r D_\ell^{-1}$$

stable. Such a  $T_0$  is given by

ORIGINAL PAGE IS  
OF POOR QUALITY

$$T_o = \begin{bmatrix} 0 & 0 \\ 0 & \frac{9}{(s-2)^2} \end{bmatrix}$$

ORIGINAL PAGE IS  
OF POOR QUALITY

$$-N_r^{-1} T_o D_\ell^{-1} + U_r D_\ell^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}$$

We then study

$$-N_r^{-1} T_s D_\ell^{-1} = \begin{bmatrix} -T_{s1}(s+1) & T_{s2}(s+2)(s+1) \\ 0 & -T_{s2} \frac{(s+2)^2}{s-1} \end{bmatrix}$$

In order to make this term stable, we must have the following:

1. Let  $T_{s1} = n_{s1}/d_{s1}$ , so that  
 $\deg(n_{s1}) + 1 < \deg(d_{s1})$ .
2. Let  $T_{s2} = n_{s2}/d_{s2}$ , so that  
 $\deg(n_{s2}) + 2 < \deg(d_{s2})$ .
3. Zeros of  $d_{s1}$  and  $d_{s2}$  are in the open LHP.
4. Zeros of  $n_{s1}$  contain  $s = 1$ .

Based on points 1 through 4, we have the simplest

$$T_{s1} = a/(s+b)$$

and the simplest

$$T_{s2} = c(s-1)/(s+2)^2(s+d),$$

with positive  $b$  and  $d$ . For tracking step inputs, both  $1 - T_{o1} - T_{s1}$  and  $1 - T_{o2} - T_{s2}$  must have a zero at  $s = 0$ . This requires that  $b - a = 0$  and  $4d - 9d + c = 0$ . It can be seen that there are many solutions for  $a$ ,  $b$ ,  $c$  and  $d$ . Two sets of solutions are given below, together with the corresponding closed-loop transfer functions and compensators. Choosing  $b = 1$  and  $d = 1$ , we have  $a = 1$ ,  $c = 5$  and

$$T = T_o + T_s = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{14s+4}{(s+2)^2(s+1)} \end{bmatrix}$$

ORIGINAL PAGE IS  
OF POOR QUALITY

$$C = \begin{bmatrix} \frac{s+1}{s} & \frac{-(14s+4)(s-1)}{s(s+6)} \\ 0 & \frac{14s+4}{s(s+6)} \end{bmatrix}$$

Choosing  $b = 1$  and  $d = 3$ , we have  $a = 1$ ,  $c = 15$  and

$$T = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{24s+12}{(s+2)^2(s+3)} \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{s+1}{s} & \frac{-3(8s+4)(s-1)}{s(s+8)} \\ 0 & \frac{3(8s+4)}{s(s+8)} \end{bmatrix}$$

## 2.5. Hidden Modes

It is known that feedback design using transfer functions may bring about unwanted stable modes hidden in the closed-loop system (ref. 5). In the example of the previous section, the closed-loop system has a transfer function  $T(s)$  that corresponds to a fourth order system. However, the plant  $P$  is a second order system and the compensator  $C$  is a third order

system which means that the closed-loop system is actually a fifth order system. The difference in the order of the closed-loop system and its transfer function suggests that there is a hidden mode. The hidden mode in the example is at  $s = -1$  which has resulted from the cancellation of the pole of  $P$  and the zero of  $C$  at  $s = -1$ . To prevent this type of cancellation, zeros of  $I - T$  should be chosen to match the stable poles of  $P$  (which are zeros of  $P^{-1}$ ) in forming  $C$ . This selection may prevent us from choosing the simplest  $T$  in our design procedures. However, this should not be considered as a setback for finding the simplest compensators, but rather a procedure that guarantees the correct representation of a closed-loop system by its transfer function. With this procedure, the design in the example of the previous section is modified as follows.

The stable pole of  $P$  is at  $s = -1$ . This pole appears in the (1,1) element of  $P^{-1}$  as a zero. Hence,  $I - T_{o1} = T_{s1} = I - T_{s1}$  should have a zero at  $s = -1$  in addition to the zero at  $s = 0$  required for tracking step inputs. This requires that  $\deg(d_{s1}) > 2$ . The simplest  $T_{s1}$  that has this property is of the form

$$T_{s1} = \frac{cs + d}{(s+a)(s+b)}$$

ORIGINAL PAGE IS  
OF POOR QUALITY

with  $a$  and  $b > 0$ . We must have

$$1 - T_{(s)} = \frac{s^2 + (a+b)s + ab - cs - d}{(s+a)(s+b)} = \frac{s(s+1)}{(s+a)(s+b)}$$

That means  $a + b - c = 1$  and  $ab - d = 0$ . Again, there are many solutions for  $a$ ,  $b$ ,  $c$  and  $d$ . Choosing  $a = 3$  and  $b = 4$ , we have  $c = 6$ ,  $d = 12$  and

$$T_{s1} = \frac{6s+12}{(s+3)(s+4)}$$

Using a set of  $T_{o2}$  and  $T_{s2}$  as before, we have

$$T = \begin{bmatrix} \frac{6(s+2)}{(s+3)(s+4)} & 0 \\ 0 & \frac{24s+12}{(s+2)^2(s+3)} \end{bmatrix}$$

ORIGINAL PAGE IS  
OF POOR QUALITY

and

$$C(s) = \begin{bmatrix} \frac{6(s+2)}{s} & \frac{-(24s+12)(s-1)}{s(s+9)} \\ 0 & \frac{24s+12}{s(s+8)} \end{bmatrix}$$

Note that  $C$  remains a third order system and the order of  $T(s)$  is 5 which means that there is no longer a hidden mode.

### 3. EIGENVALUE/EIGENVECTOR ASSIGNMENT PROCEDURES

#### 3.1. Introduction

The design of multivariable feedback control systems using eigenvalue/eigenvector assignment procedures has received considerable attention during the past several years. Several early studies (refs. 11, 12) focused on an algebraic formulation of the spectral assignment problem. More recent studies (refs. 13-15) have been successful in developing a geometric formulation of this problem. In (ref. 13) the total design freedom available to assign eigenvectors is characterized in terms of eigenspaces. The use of this freedom to achieve desired design specifications has been the subject of an extensive investigation by the current authors and colleagues.

Procedures have been developed for approximating desired mode mixing (ref. 16), reducing eigensystem sensitivity to variations in plant parameters (refs. 17, 18), and reducing the effects of actuator noise on a statistical measure of system performance (ref. 19, 20). In addition, a procedure for modifying the feedback gain matrix to satisfy specified gain constraints (ref. 21, 22) while maintaining a given mode mix has been devised. More recently these procedures have been combined into a single unified design philosophy (ref. 7). This philosophy is reviewed and a computer aided design software package to implement the design philosophy is presented in this section.



### 3.2. Design Philosophy

The new eigenvalue/eigenvector assignment design philosophy is illustrated in figure 2. The philosophy is based on the premise that achieving a specified set of eigenvalues and approximating a desired set of eigenvectors is of primary importance. Sensitivity reduction, noise suppression, and gain modification are assigned secondary importance and are carried out so as to preserve an initial spectral assignment.

The procedure assumes that the designer is able to identify a desired set of eigenvalues and an approximate set of desired eigenvectors. Eigenvalues directly control the rates of response of the system modes while eigenvectors control how the modes mix among the system states and/or outputs. The design begins with the specification of a desired set of eigenvalues. The procedure realizes arbitrarily specified sets of eigenvalues if the system is controllable. The specified eigenvalues are used to compute the system eigenspaces--the vector spaces in which all realizable system eigenvectors must be contained. These spaces explicitly display the total design freedom available in assigning eigenvectors for a given eigenvalue assignment. Next, the desired set of eigenvectors are specified and projected onto the eigenspaces to locate the set of realizable eigenvectors as close as possible in a minimum square error sense to the desired set of eigenvectors. Since the major advantage of the eigenvalue/eigenvector assignment procedure is the ability to assign eigenvectors, great importance is given to remaining in a small neighborhood of the initial eigenvector assignment.

After the specified eigenvalues have been assigned and the specified eigenvectors have been approximated, the resultant closed-loop system is investigated to determine if all eigenvector components are satisfactory, eigensystem sensitivity is sufficiently low, and gain magnitudes meet specified design constraints. If modification is required, new eigenvectors are selected in a manner to achieve the desired objective using a gradient search procedure. However, the gradient search is conducted local to the initial eigenvector assignment so that desired mode mixing is retained.

ORIGINAL PAGE IS  
OF POOR QUALITY

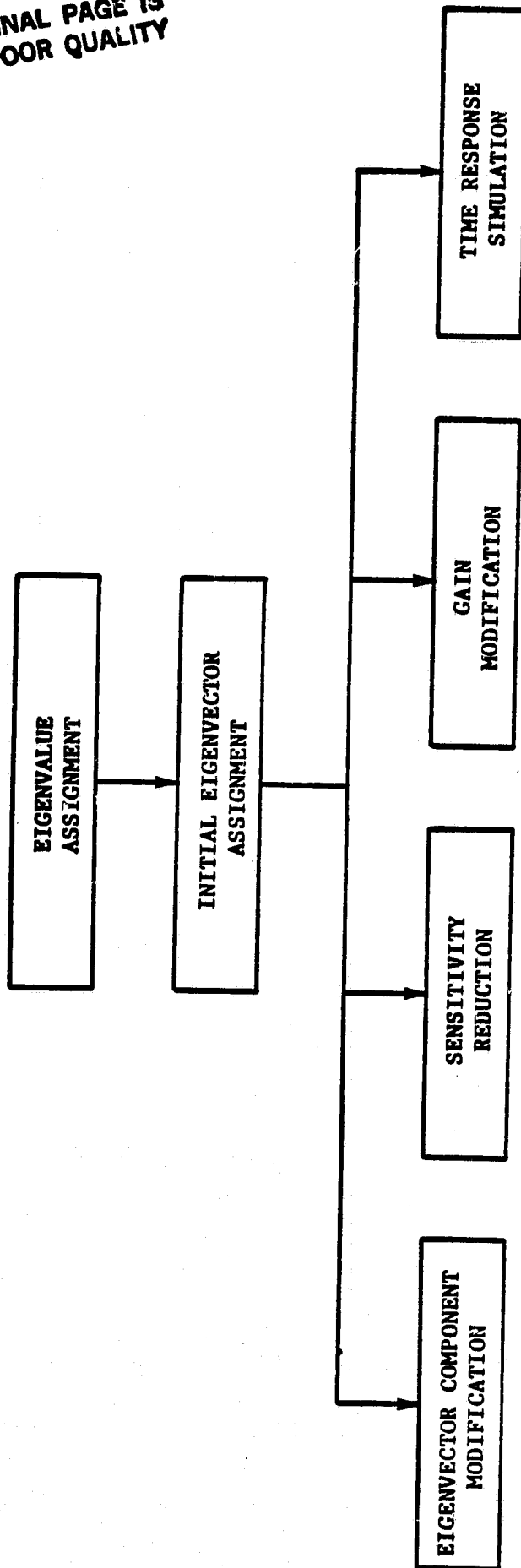


Figure 2. Eigenvalue/eigenvector assignment design philosophy.

### 3.3. Computer Aided Design Software Package

A flowchart diagram illustrating the organization of the computer software package to implement the eigenvalue/eigenvector assignment procedure is shown in figure 3. The package consists of a number of special purpose subprograms accessible from a main control program. The subprograms can be called in any order to implement specific design objectives, as shown in figure 2. The program is self-instructed and requires no familiarity on the part of the user with the mathematics of spectral assignment.

In the following, the various modes of operation of the program are discussed. An example illustrating the use of the program is presented in the next section, and the program listing is included in the Appendix.

#### Mode 0

Mode 0 provides a list of references detailing program operation.

#### Mode 1

Mode 1 is the mode in which system data is entered to the program. Required data includes the number of system states, inputs, and outputs, and the system state variable description in matrix form given by the triple (A, B, C). The user can also set the number of significant digits in user-computer communication as well as the program value for "zero."

#### Mode 2

In Mode 2, the user specifies desired closed loop system eigenvalues. This mode then internally calculates the corresponding eigenspaces for transmission to other subprograms. User selected eigenvalues are always achieved in this design procedure.

#### Mode 3

The user specifies desired eigenvectors in Mode 3. The program responds with the set of actual closed loop system eigenvectors which are

ORIGINAL PAGE IS  
OF POOR QUALITY

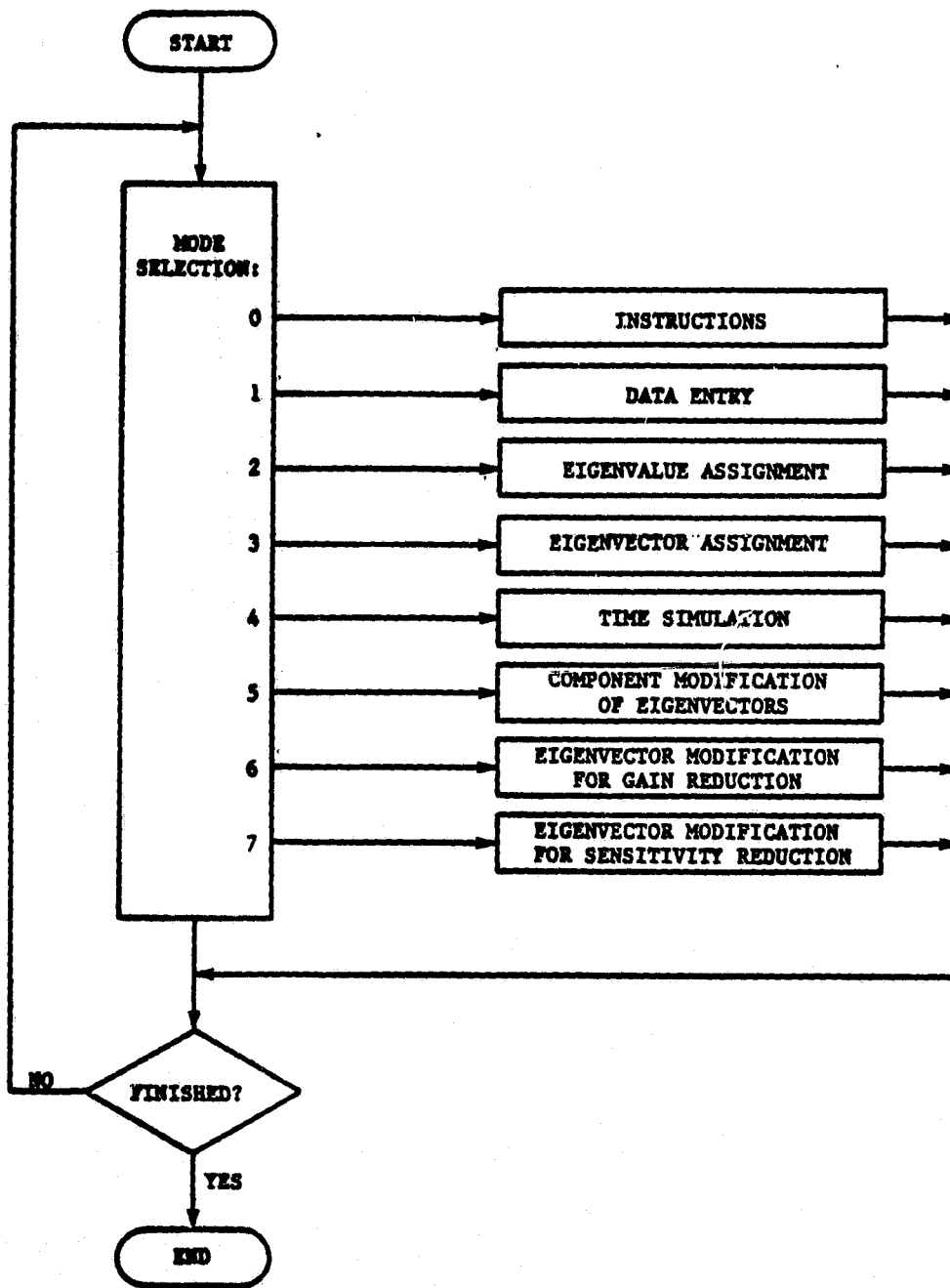


Figure 3. Spectral assignment computer software package organization.

closest to those specified in a least square error sense. The program also displays the error magnitude between each desired and realized eigenvector.

#### Mode 4

Mode 4 is a closed loop system simulation package. The subprogram numerically solves the set of system state equations subject to user specified inputs and initial conditions. Time responses are plotted separately or on the same axes for comparison.

#### Mode 5

Mode 5 allows the user to modify specified components in the eigenvector matrix while retaining current values of other components. The modification is automatically carried out using a gradient search procedure under the control of the user.

#### Mode 6

Mode 6 allows the user to modify selected components of the feedback gain matrix while maintaining an approximation to a specified eigenvector matrix. Components of the feedback matrix to be reduced are identified by row and column number. Unequal priority in reducing component magnitudes can be assigned. The modification is automatically conducted by a gradient search algorithm under the control of the user.

#### Mode 7

In Mode 7 the user can reduce eigensystem sensitivity to variations in plant parameters. The procedure utilizes a gradient search procedure to modify system closed loop eigenvectors to reduce the sensitivity of eigenvalues and eigenvectors to changes in specified components of the system state matrices.

### 3.4. Design Example

In this section an example is presented to illustrate the designer - machine dialog during the design process. Mode 1 is first entered and important system data is input.

Number of states: 3  
Number of inputs: 2  
Number of outputs: 3  
Significant digits: 6  
Program zero: 0.0001

ORIGINAL PAGE IS  
OF POOR QUALITY

$$A = \begin{bmatrix} -2.00 & 0.00 & 1.00 \\ 0.00 & -2.00 & 1.00 \\ 1.00 & 1.00 & -2.00 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 1.00 \\ 0.00 & 0.00 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

Next Mode 2 is entered and desired closed-loop system eigenvalues are input.

$$\lambda_1 = -1.00$$

$$\lambda_2 = -1.20$$

$$\lambda_3 = -3.00$$

Mode 3 is entered next and desired closed-loop system eigenvectors are input. The program responds with the actual set of eigenvectors as close as possible in a least square error sense to those specified. The program also generates the feedback matrix F which assign these eigenvectors and the specified eigenvalues.

ORIGINAL PAGE IS  
OF POOR QUALITY

$$V_{\text{desired}} = \begin{bmatrix} 3.75 & -0.67 & 1.00 \\ 3.25 & 0.75 & -1.00 \\ 7.00 & 0.00 & 0.10 \end{bmatrix}$$

$$V_{\text{actual}} = \begin{bmatrix} 3.75 & -0.70 & 0.97 \\ 3.25 & 0.72 & -1.03 \\ 7.00 & 0.02 & 0.07 \end{bmatrix}$$

$$F = \begin{bmatrix} 13.25 & 12.53 & -13.38 \\ -13.16 & -12.45 & 12.30 \end{bmatrix}$$

The designer then enters Mode 4 to simulate the closed-loop system just designed. The user specifies initial conditions and system inputs.

$$x(0) = \begin{bmatrix} 1.00 \\ 0.00 \\ 0.00 \end{bmatrix}$$

$$u(t) = \begin{bmatrix} 0.00 \\ 1.00 \end{bmatrix}$$

The program responds with plots of the system inputs and states shown as functions of time. Plot are also shown for another set of initial conditions and zero input.

$$x(0) = \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \end{bmatrix}$$

ORIGINAL PAGE IS  
OF POOR QUALITY

$$u(t) = \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}$$

It is demonstrated that curves may be requested separately or together for comparison. The designer next enters Mode 5 to modify a component of the eigenvector matrix. The designer specifies that he desires to reduce the magnitude of the (3,1) element of V. Equal weight is assigned to reducing this component and to retaining the current values of other components. After three iterations, a satisfactory V is obtained. The program displays the new feedback gain matrix for this assignment.

$$V = \begin{bmatrix} 3.18 & -0.70 & 0.97 \\ 2.68 & 0.72 & -1.03 \\ 5.85 & 0.02 & 0.07 \end{bmatrix}$$

$$F = \begin{bmatrix} 13.25 & 12.54 & -13.38 \\ -13.16 & -12.45 & 12.29 \end{bmatrix}$$

The designer then returns to Mode 4 to again display the system states.

$$x(0) = \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \end{bmatrix}$$

$$u(t) = \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}$$

Finally, the designer enters Mode 6 to attempt to reduce the magnitudes of entries in the gain matrix without greatly changing the eigenvector assignment. Equal weight is placed on reducing each component of F. After three iterations a new V matrix and corresponding F matrix are obtained.



ORIGINAL PAGE IS  
OF POOR QUALITY

$$V = \begin{bmatrix} 3.18 & -0.66 & 0.75 \\ 2.68 & 0.76 & -1.24 \\ 5.85 & 0.16 & 0.48 \end{bmatrix}$$

$$F = \begin{bmatrix} 2.21 & 1.44 & -2.31 \\ -2.12 & -1.41 & 1.25 \end{bmatrix}$$

Not demonstrated here but included in the program is a sensitivity reduction mode. The program is also fully capable of dealing with complex eigenvalue and eigenvector assignments.

\*\*\*\*\*  
\*\*\*\*\* SPECTRAL ASSIGNMENT PACKAGE \*\*\*\*\*

ENTER DESIRED MODE OF OPERATION, MODE=0,1,2,...,8:

1  
\*\*\*\*\* MODE 1: DATA ENTRY \*\*\*\*\*

\*\*\*\*\* ENTER OR CHANGE SYSTEM PARAMETERS:

PREVIOUS VALUES?

1  
NS= 3            NI= 2            NU= 3            LDGT= 6            ZERO= 0.00001000000

WISH TO CHANGE?

0  
MATRIX A :  
          1                    2                    3  
1 -0.200000E+01    0.000000E+00    0.100000E+01  
2  0.000000E+00    -0.200000E+01    0.100000E+01  
3  0.100000E+01    0.100000E+01    -0.200000E+01

WISH TO CHANGE?

0  
MATRIX B :  
          1                    2  
1  0.100000E+01    0.000000E+00  
2  0.000000E+00    0.000000E+00  
3  0.000000E+00    0.000000E+00

ORIGINAL PAGE IS  
OF POOR QUALITY

WISH TO CHANGE?

1  
ENTER NEW VALUE(S) :

1.000  0.000  
0.000  1.000  
0.000  0.000

MATRIX C :  
          1                    2                    3  
1  0.100000E+01    0.000000E+00    0.000000E+00  
2  0.100000E+01    0.000000E+00    0.000000E+00  
3  0.000000E+00    0.100000E+01    0.100000E+01

WISH TO CHANGE?

1  
ENTER NEW VALUE(S) :

1.000  0.000  0.000  
0.000  1.000  0.000  
0.000  0.000  1.000

WISH TO EXIT FROM THIS MODE?

1  
\*\*\*\*\* EXITING MODE 1 \*\*\*\*\*  
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?

0  
\*\*\*\*\*  
\*\*\*\*\* SPECTRAL ASSIGNMENT PACKAGE \*\*\*\*\*

ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:

2

\*\*\*\*\* MODE 2:EIGENVALUE ASSIGNMENT \*\*\*\*\*

\*\*\*\*\* ENTER OR CHANGE EIGENVALUES:

PREVIOUS VALUES?

0

LAMBDA 1:

REAL= 0.000000E+00 IMAG= 0.000000E+00

WISH TO CHANGE?

1

enter new value(s) :

-1.000 0.000

NEXT EIGENVALUE:

PREVIOUS VALUES?

0

LAMBDA 2:

REAL= 0.000000E+00 IMAG= 0.000000E+00

WISH TO CHANGE?

1

enter new value(s) :

-1.2000 0.0000

NEXT EIGENVALUE:

PREVIOUS VALUES?

1

LAMBDA 3:

REAL= -0.300000E+01 IMAG= 0.000000E+00

WISH TO CHANGE?

0

WISH TO EXIT FROM THIS MODE?

1

\*\*\*\*\* EXITING MODE 2 \*\*\*\*\*  
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?

0

\*\*\*\*\* SPECTRAL ASSIGNMENT PACKAGE \*\*\*\*\*

ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:

3

\*\*\*\*\* MODE 3:EIGENVECTOR ASSIGNMENT \*\*\*\*\*

\*\*\*\*\* ENTER OR CHANGE EIGENVECTORS:

PREVIOUS VALUES?

0

EIGENVECTOR V 1:

(REAL)	(IMAG)
0.000000E+00	0.000000E+00
0.000000E+00	0.000000E+00
0.000000E+00	0.000000E+00

WISH TO CHANGE?

1

ORIGINAL PAGE 13  
OF POOR QUALITY

ENTER A NEW DESIRED VECTOR :

3.75 0.00

3.25 0.00

7.00 0.00

DESIRED VECTOR:

0.375000E+01 0.325000E+01 0.700000E+01  
ACTUAL VECTOR:

0.375000E+01 0.325000E+01 0.700000E+01  
ERROR VECTOR:

0.298023E-07 0.298023E-07 0.000000E+00  
LENGTH OF THE DESIRED VECTOR = 8.580501  
LENGTH OF THE PROJECTED VECTOR = 8.580501  
LENGTH OF THE ERROR VECTOR = 0.000000  
IS THE ERROR ACCEPTABLE?

1

NEXT EIGENVECTOR:

EIGENVECTOR V 2: (REAL) (IMAG)  
0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00

WISH TO CHANGE?

1

ENTER A NEW DESIRED VECTOR :

-.6700 0.000

.75000 0.000

0.000 0.00

DESIRED VECTOR:

-0.670000E+00 0.750000E+00 0.000000E+00  
ACTUAL VECTOR:

-0.700303E+00 0.719697E+00 0.242424E-01  
ERROR VECTOR:

0.303030E-01 0.303030E-01 -0.242424E-01  
LENGTH OF THE DESIRED VECTOR = 1.005684  
LENGTH OF THE PROJECTED VECTOR = 1.004478  
LENGTH OF THE ERROR VECTOR = 0.049237  
IS THE ERROR ACCEPTABLE?

1

NEXT EIGENVECTOR:

EIGENVECTOR V 3: (REAL) (IMAG)  
0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00

WISH TO CHANGE?

1

ENTER A NEW DESIRED VECTOR :

1.000 0.000

-1.000 0.000

.1000 0.000

DESIRED VECTOR:

0.100000E+01 -0.100000E+01 0.100000E+00  
ACTUAL VECTOR:

ORIGINAL PAGE IS  
OF POOR QUALITY

0.966667E+00 -0.103333E+01 0.666667E-01  
ERROR VECTOR:

0.333333E-01 0.333333E-01 0.333333E-01  
LENGTH OF THE DESIRED VECTOR = 1.417745  
LENGTH OF THE PROJECTED VECTOR= 1.416569  
LENGTH OF THE ERROR VECTOR = 0.057735  
IS THE ERROR ACCEPTABLE?

1  
-----CONTENTS OF 'CURRNT' DATA FILE INCLUDE:  
MATRIX U :

	1	2	3
1	0.375000E+01	-0.700303E+00	0.966667E+00
2	0.325000E+01	0.719697E+00	-0.103333E+01
3	0.700000E+01	0.242424E-01	0.666667E-01

WISH TO DISPLAY THE NORMALIZED EIGENVECTORS?

1  
NORMALIZED VECTORS :

	1	2	3
1	0.437037E+00	-0.697181E+00	0.682400E+00
2	0.378766E+00	0.716489E+00	-0.729462E+00
3	0.815803E+00	0.241344E-01	0.470621E-01

GAIN MATRIX F:

	1	2	3
1	0.132526E+02	0.125341E+02	-0.133833E+02
2	-0.131593E+02	-0.124526E+02	0.122955E+02

MATRIX AHAT:

	1	2	3
1	0.112526E+02	0.125341E+02	-0.123833E+02
2	-0.131593E+02	-0.144526E+02	0.132955E+02
3	0.100000E+01	0.100000E+01	-0.200000E+01

WISH TO EXIT FROM THIS MODE?

1  
\*\*\*\*\* EXITING MODE 3 \*\*\*\*\*  
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?

0  
\*\*\*\*\* SPECTRAL ASSIGNMENT PACKAGE \*\*\*\*\*

ENTER DESIRED MODE OF OPERATION, MODE=0,1,2,...,9:

4

\*\*\*\*\* MODE 4: TIME SIMULATION \*\*\*\*\*

\*\*\*\*\* CHOOSE SIMULATION OPTIONS:

-ENTER: 1 TO SIMULATE [A], 2 TO SIMULATE [AHAT], (3 FOR [ATIL]):

2

ENTER 0 TO SIMULATE OUTPUTS, 1 TO SIMULATE STATE VARIABLES:

1

ENTER SIMULATION TIME, (REAL NUMBER IN SECONDS):

5

ENTER NUMBER OF POINTS TO BE CALCULATED, (200 MAX):

150

SPECIFY THE INITIAL CONDITIONS:

X 1(0):

1

X 2(0):

0

X 3(0):

0

CHOOSE INPUT OPTIONS: 1 FOR NO INPUT, 2 FOR A STEP INPUT,  
3 FOR A RAMP, AND 4 FOR A TRUNCATED RAMP:

INPUT OPTION FOR U 1:

1

INPUT OPTION FOR U 2:

2

SPECIFY AMPLITUDE OF THE STEP INPUT U 2:

1

ENTER 0 FOR 80 DISPLAY COLUMNS, 1 FOR 129 COLUMNS:

0

ENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:

0

DO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?

0

POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER  
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)

1 HERE WE GO

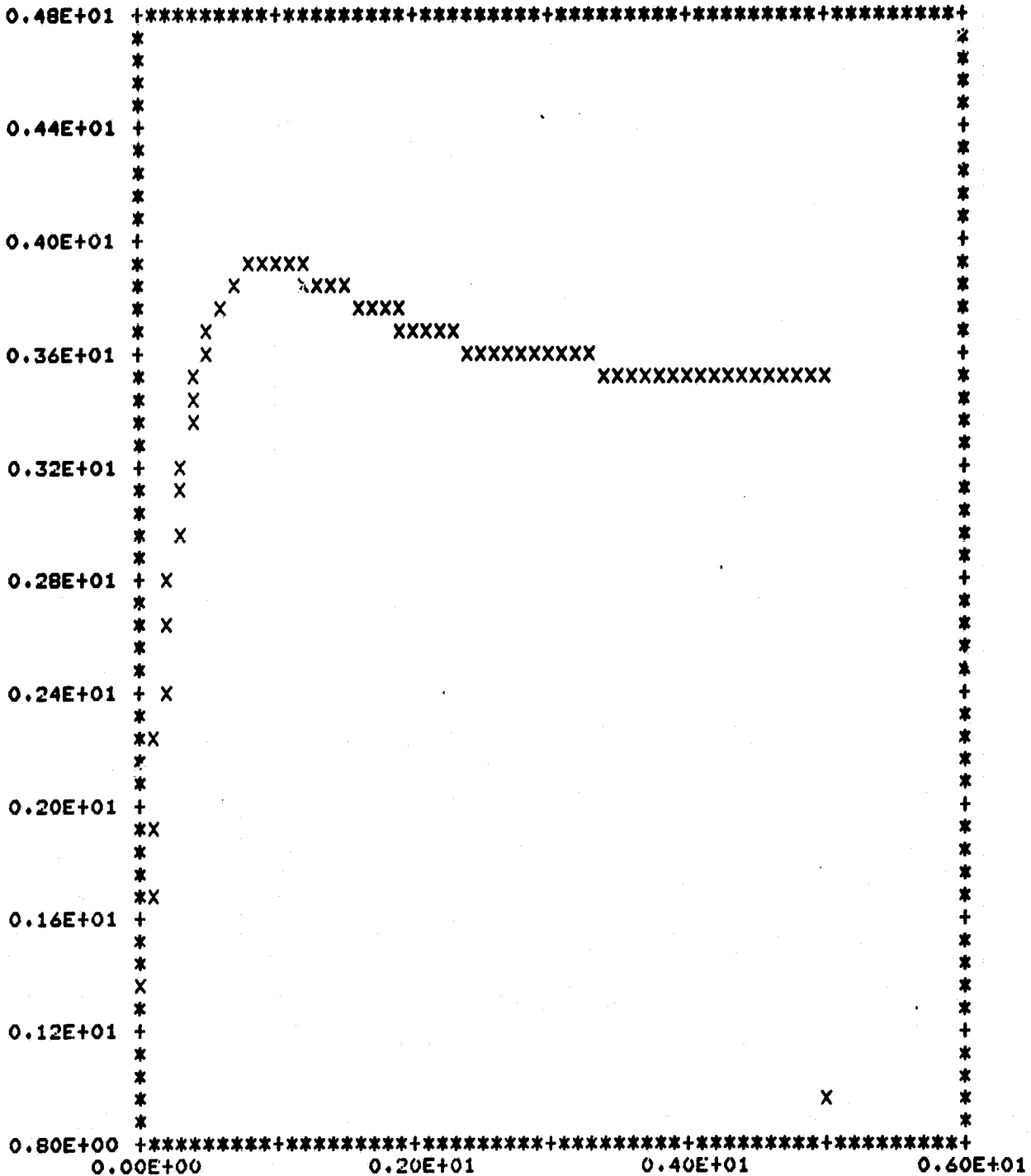
ORIGINAL PAGE IS  
OF POOR QUALITY



TIME SIMULATION

ORIGINAL PAGE IS  
OF POOR QUALITY

R  
E  
S  
P  
O  
N  
S  
E



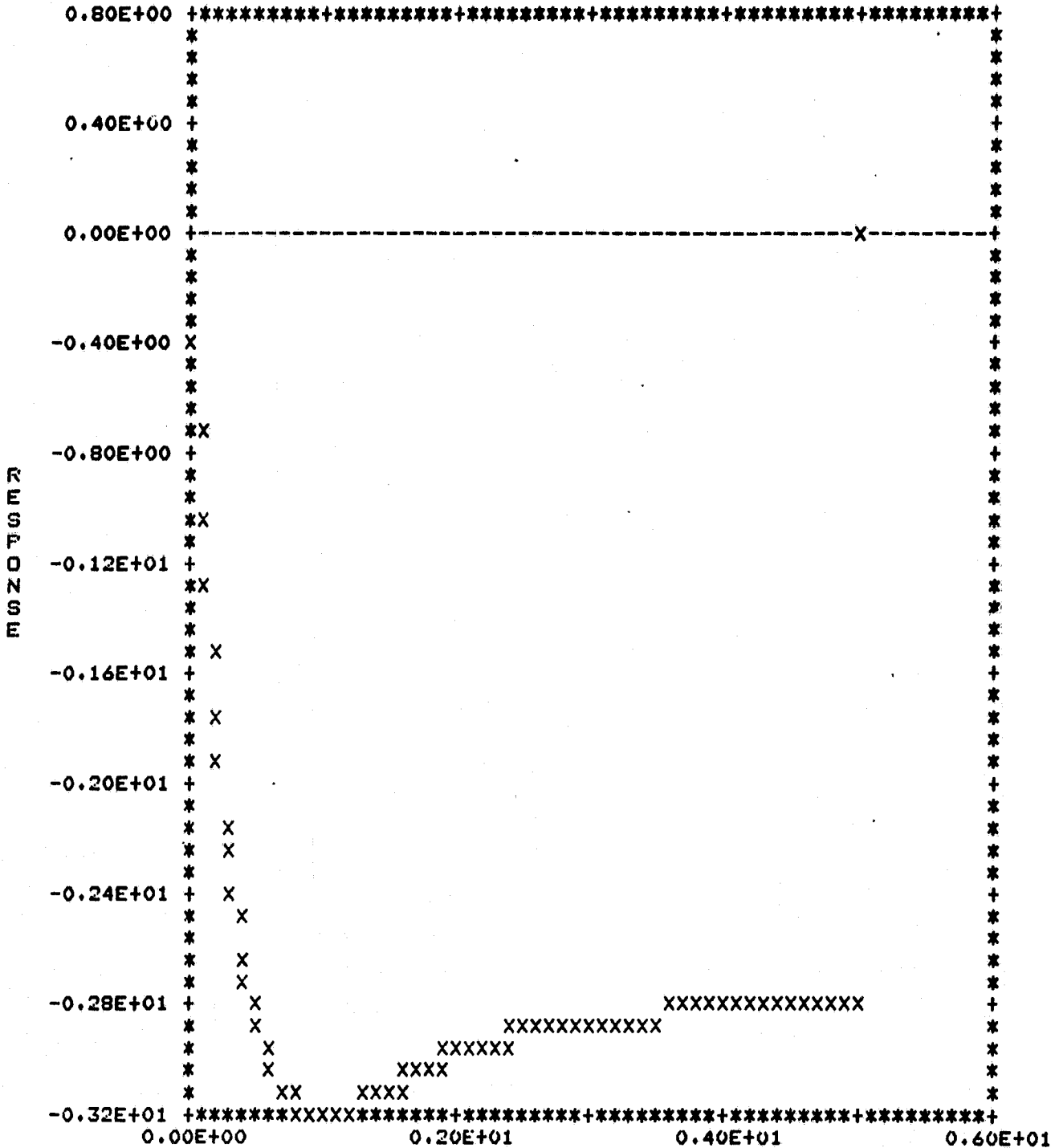
T I M E

POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER  
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)

1



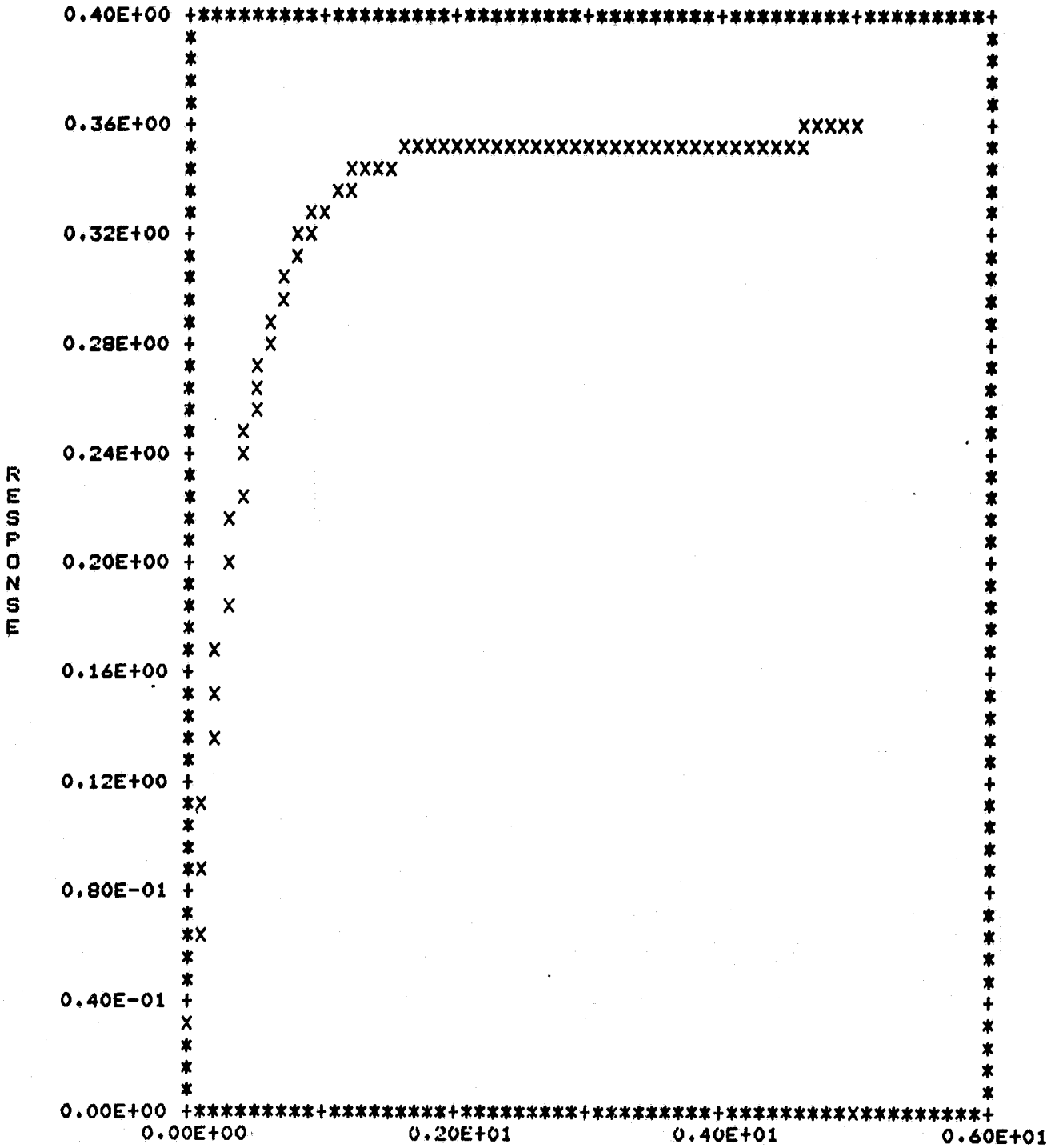
TIME SIMULATION



POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER  
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)

1

TIME SIMULATION



WISH TO REPEAT THE PLOTTING?  
0  
WISH TO EXIT FROM THIS MODE?  
0

ORIGINAL PAGE IS  
OF POOR QUALITY

\*\*\*\*\* MODE 4: TIME SIMULATION \*\*\*\*\*

\*\*\*\*\* CHOOSE SIMULATION OPTIONS:

-ENTER: 1 TO SIMULATE [A], 2 TO SIMULATE [AHAT], (3 FOR [ATIL]):

2  
ENTER 0 TO SIMULATE OUTPUTS, 1 TO SIMULATE STATE VARIABLES:

1  
ENTER SIMULATION TIME, (REAL NUMBER IN SECONDS):

5  
ENTER NUMBER OF POINTS TO BE CALCULATED, (200 MAX):

150  
SPECIFY THE INITIAL CONDITIONS:

X 1(0):

1

X 2(0):

1

X 3(0):

1

CHOOSE INPUT OPTIONS: 1 FOR NO INPUT, 2 FOR A STEP INPUT,  
3 FOR A RAMP, AND 4 FOR A TRUNCATED RAMP:

INPUT OPTION FOR U 1:

1

INPUT OPTION FOR U 2:

1

ENTER 0 FOR 80 DISPLAY COLUMNS, 1 FOR 129 COLUMNS:

0

ENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:

1

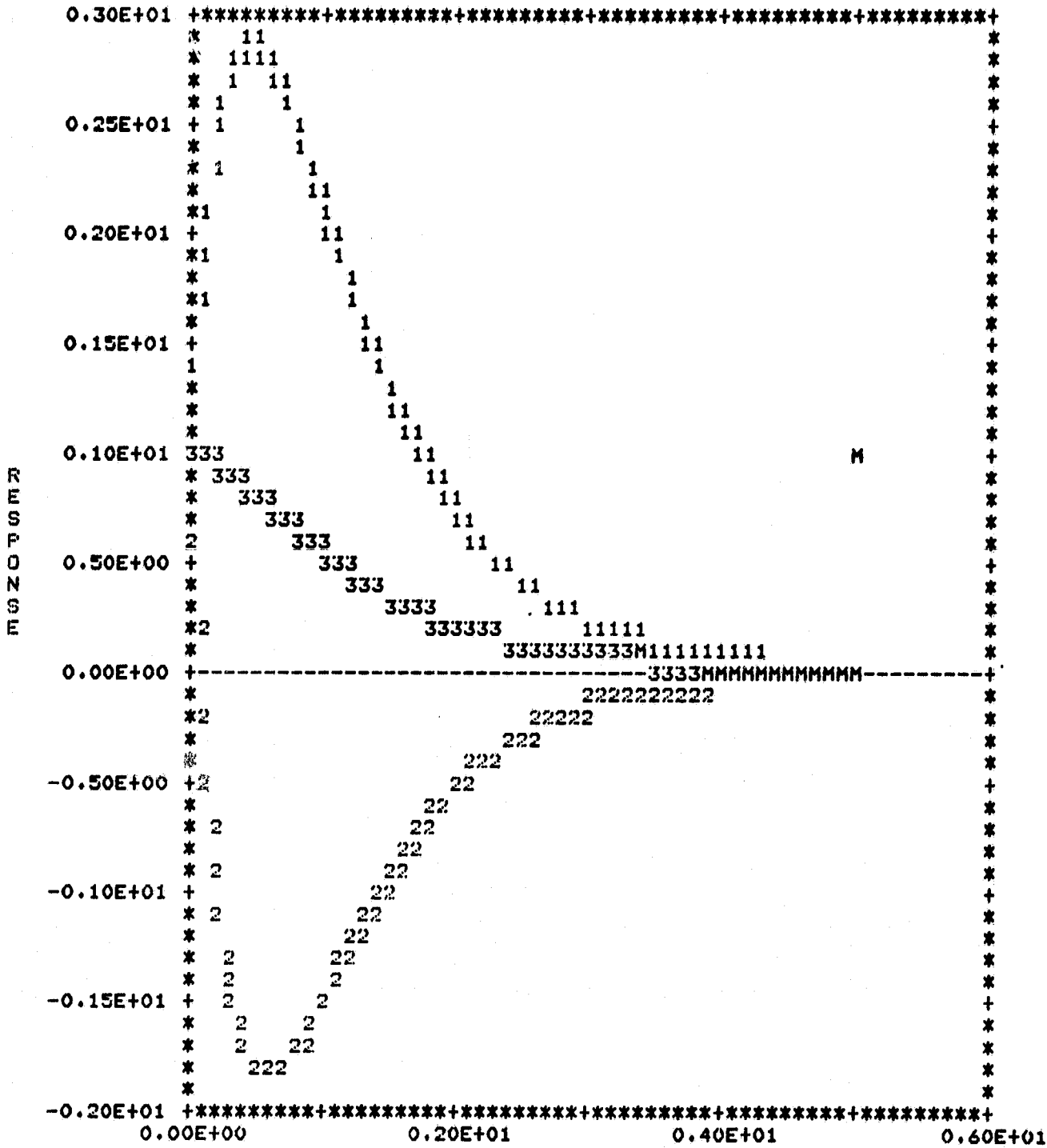
DO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?

0

POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER  
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)

L 1

TIME SIMULATION



T I M E

WISH TO REPEAT THE PLOTTING?  
[SYSTEM] This Job will be killed in 4 minutes if it remains inactive  
0  
WISH TO EXIT FROM THIS MODE?   
1  
\*\*\*\*\* EXITING MODE 4 \*\*\*\*\*

ORIGINAL PAGE IS  
OF POOR QUALITY

\*\*\*\*\*  
\*\*\*\*\* SPECTRAL ASSIGNMENT PACKAGE \*\*\*\*\*

ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:

5

\*\*\*\*\* MODE 5: COMPONENT REDUCTION \*\*\*\*\*

ENTER THE COORDINATES OF THE COMPONENT TO BE REDUCED  
ROW=--, COLUMN=--(BOTH INTEGERS):

3 1

SET DESIRED WEIGHTS,DEFAULT VALUES ARE:

F1=F2=1.000

WISH TO CHANGE?

0

J1= 0.490000E+02 J2= 0.000000E+00

COST= 0.490000E+02

GRADIENT MATRIX:

1 2 3

1 -0.707107E+00 0.000000E+00 0.000000E+00

2 -0.707107E+00 0.000000E+00 0.000000E+00

GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

Default values are:

# of steps,N= 1 step size,d= 0.100000E-01 dmin= 0.100000E-04

Wish to change?

0

J1= 0.488022E+02 J2= 0.999999E-04

NEW COST= 0.488023E+02

Cost Function= 0.488023E+02

Wish to continue the search?

1

GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

Default values are:

# of steps,N= 1 step size,d= 0.100000E-01 dmin= 0.100000E-04

Wish to change?

1

Enter new values:

1 0.5 .0001

J1= 0.394227E+02 J2= 0.260100E+00

NEW COST= 0.396828E+02

Cost Function= 0.396828E+02

Wish to continue the search?

1

GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

Default values are:

# of steps,N= 1 step size,d= 0.500000E+00 dmin= 0.100000E-03

Wish to change?

1

Enter new values:

1 .3 .0001

ORIGINAL PAGE 13  
OF POOR QUALITY

J1= 0.342750E+02 J2= 0.656100E+00  
NEW COST= 0.349311E+02

Cost Function= 0.349311E+02  
Wish to continue the search?

0  
MATRIX V :

	1	2	3
1	0.317724E+01	-0.700303E+00	0.966667E+00
2	0.267724E+01	0.719697E+00	-0.103333E+01
3	0.585449E+01	0.242424E-01	0.666667E-01

WISH TO DISPLAY THE NORMALIZED EIGENVECTORS?

0  
GAIN MATRIX F:

	1	2	3
1	0.132540E+02	0.125355E+02	-0.133827E+02
2	-0.131608E+02	-0.124540E+02	0.122949E+02

TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?

1

\*\*\*\*\* MODE 4: TIME SIMULATION \*\*\*\*\*

\*\*\*\*\* CHOOSE SIMULATION OPTIONS:

-ENTER: 1 TO SIMULATE [A], 2 TO SIMULATE [AHAT], (3 FOR [ATIL]):

2

ENTER 0 TO SIMULATE OUTPUTS, 1 TO SIMULATE STATE VARIABLES:

1

ENTER SIMULATION TIME, (REAL NUMBER IN SECONDS):

5

ENTER NUMBER OF POINTS TO BE CALCULATED, (200 MAX):

150

SPECIFY THE INITIAL CONDITIONS:

X 1(0):

1

X 2(0):

1

X 3(0):

1

CHOOSE INPUT OPTIONS: 1 FOR NO INPUT, 2 FOR A STEP INPUT,  
3 FOR A RAMP, AND 4 FOR A TRUNCATED RAMP:

INPUT OPTION FOR U 1:

1

INPUT OPTION FOR U 2:

1

ENTER 0 FOR 80 DISPLAY COLUMNS, 1 FOR 129 COLUMNS:

0

ENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:

1

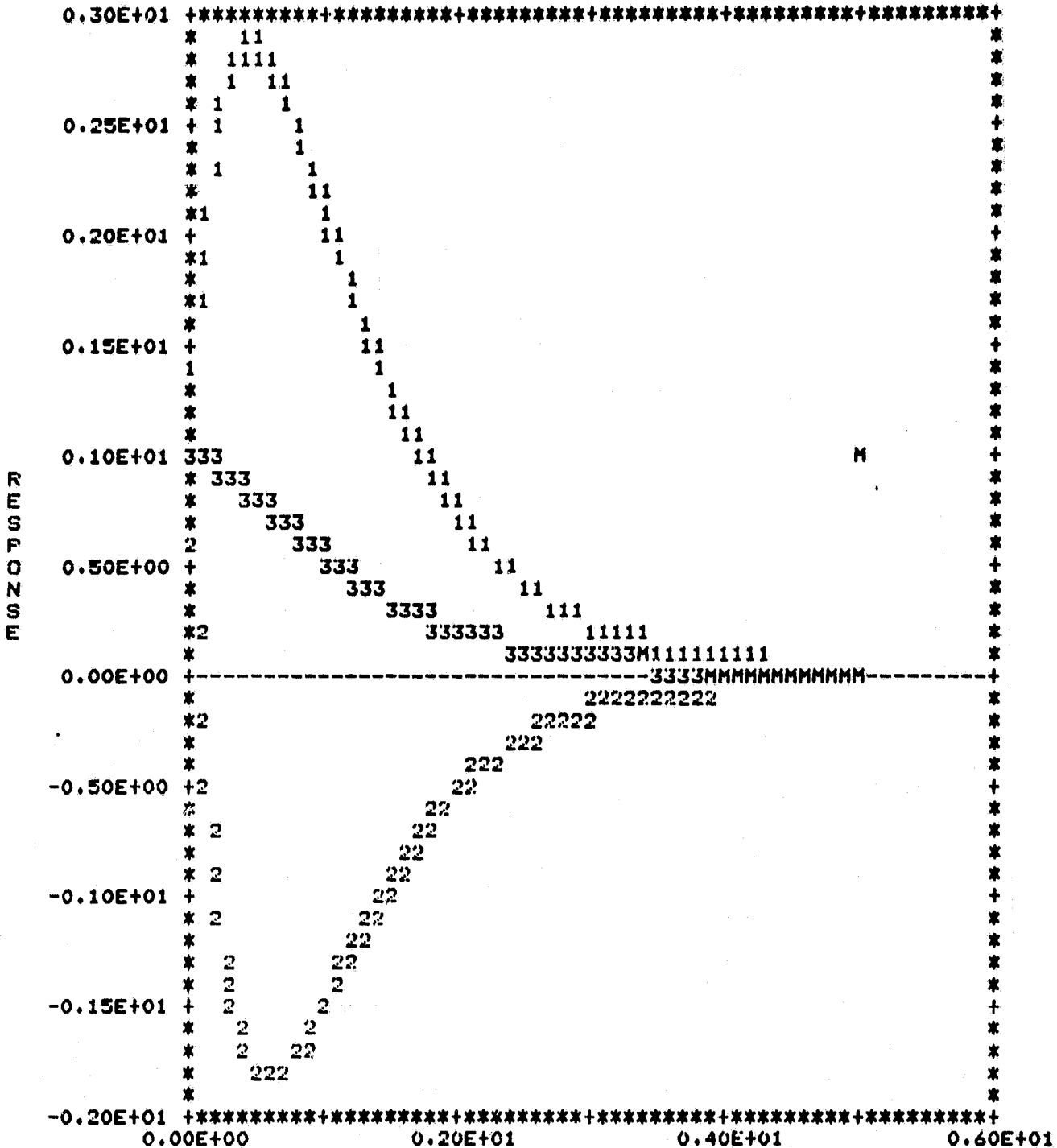
DO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?

0

POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER  
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)

1

TIME SIMULATION



WISH TO REPEAT THE PLOTTING?

0\0\0

WISH TO EXIT FROM THIS MODE?

1

\*\*\*\*\* EXITING MODE 4 \*\*\*\*\*

ORIGINAL PAGE IS  
OF POOR QUALITY

\*\*\*\*\*  
\*\*\*\*\* SPECTRAL ASSIGNMENT PACKAGE \*\*\*\*\*

ENTER DESIRED MODE OF OPERATION, MODE=0,1,2,...,8:

6

MATRIX V :

	1	2	3
1	0.317724E+01	-0.700303E+00	0.966667E+00
2	0.267724E+01	0.719697E+00	-0.103333E+01
3	0.585449E+01	0.242424E-01	0.666667E-01

GAIN MATRIX F:

	1	2	3
1	0.132540E+02	0.125355E+02	-0.133827E+02
2	-0.131608E+02	-0.124540E+02	0.122949E+02

\*\*\*\*\* MODE 6: GAIN REDUCTION \*\*\*\*\*

SET ALPHA PARAMETERS :

DEFAULT VALUES ARE :

GAIN PARAMETERS :

	1	2	3
1	0.100000E+01	0.100000E+01	0.100000E+01
2	0.100000E+01	0.100000E+01	0.100000E+01

WISH TO CHANGE:

0

COST= 0.991380E+03

Gradient matrix:

	1	2	3
1	-0.328984E-04	0.123434E+00	-0.719359E+00
2	0.390425E-04	0.120108E+00	-0.672949E+00

GRADIENT SEARCH ROUTINE, SET SEARCH PARAMETERS:

Default values are:

# of steps, N= 1    step size, d= 0.100000E-01    dmin= 0.100000E-04

Wish to change?

1

Enter new values:

3

.1 .0001

NEW COST= 0.107344E+03

NEW COST= 0.389868E+02

NEW COST= 0.203202E+02

Cost Function= 0.203202E+02

Wish to continue the search?

0

MATRIX V :

	1	2	3
1	0.317723E+01	-0.663273E+00	0.750859E+00
2	0.267726E+01	0.755729E+00	-0.123522E+01



ORIGINAL PAGE IS  
OF POOR QUALITY

3 0.585449E+01 0.115571E+00 0.484359E+00  
WISH TO DISPLAY THE NORMALIZED EIGENVECTORS?

0

GAIN MATRIX F:

1

2

3

1 0.220673E+01 0.143515E+01 -0.231119E+01

2 -0.212192E+01 -0.140673E+01 0.125216E+01  
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?

1

## REFERENCES

1. D. C. Youla, J. J. Bongiorno, Jr., and H.A. Jabr, "Modern Wiener-Hopf Design of Optimal Controllers - Part I: The Single-Input-Output Case." IEEE Trans. on Automatic Control, Vol. AC-21, February 1976.
2. D. C. Youla, H. A. Jabr, and J. J. Bongiorno, Jr., "Modern Wiener-Hopf Design of Optimal Controllers - Part II: The Multivariable Case." IEEE Trans. on Automatic Control, Vol. AC-21, June 1976.
3. C. A. Desoer, R. W. Lieu, J. Murray, and R. Saeks, "Feedback System Design: The Fractional Representation Approach to Analysis and Synthesis." IEEE Trans. on Automatic Control, Vol. AC-25, June 1980.
4. J. L. Peczkowski and M. K. Sain, "Linear Multivariable Synthesis with Transfer Functions," in Alternatives for Linear Multivariable Control, edited by Sain, Peczkowski and Melsa, NEC, Chicago, 1978.
5. P. J. Antsaklis and M. K. Sain, "Unity Feedback Compensation of Unstable Plants." Proceedings of IEEE Conference on Decision and Control, December 1981.
6. R. R. Mielke and L. J. Tung, "Design of Multivariable Feedback Control System Via Spectral Assignment." Progress Report, NASA Grant NSG-1650, Old Dominion University Research Foundation, October 1981.
7. R. R. Mielke, L. J. Tung and M. Marefat, "Design of Multivariable Feedback Control System Via Spectral Assignment." Progress Report, NASA Grant NSG-1650, Old Dominion University Research Foundation, June 1982.
8. R. Saeks and J. Murray, "Feedback System Design: The Tracking and Disturbance Rejection Problems." IEEE Transactions on Automatic Control, Vol. AC-26, February 1981.
9. R. R. Mielke, L. J. Tung, and S. R. Liberty, "Design of Multivariable Feedback Control Systems Via Spectral Assignment." Progress Report NASA Grant NSG-1650, Old Dominion University Research Foundation, April 1981.
10. F. M. Brasch, Jr., and J. B. Pearson, "Pole Placement Using Dynamic Compensators.", IEEE Trans. on Automatic Control, Vol. AC-15, No. 1, February 1970.
11. S. Srinathkumar: "Spectral Characterization of Multi-Input Dynamic Systems." Ph.D. dissertation, Oklahoma State University, Stillwater, OK, May 1976.
12. S. Srinathkumar: "Eigenvalue/Eigenvector Assignment Using Output Feedback." IEEE Transactions on Automatic Control, Vol. AC-23, February 1978, pp. 79-81.
13. B.C. Moore: "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed-Loop Eigenvalue Assignment." IEEE Transactions on Automatic Control, Vol. AC-21, October 1976, pp. 685-692.

14. H. Kumura: "A Further Result on the Problem of Pole Assignment by Output Feedback." IEEE Transactions on Automatic Control, Vol. AC-22, June 1977, pp. 458-463.
15. E. J. Davison and S. H. Wang: "On POle Assignment in Linear Multivariable Systems." IEEE Transactions on Automatic Control, Vol. AC-20, August 1975, pp. 516-518.
16. R. A. Maynard, R. R. Mielke, S. R. Liberty, and S. Srinathkumar: "A New Spectral Synthesis Procedure for Multivariable Regulators." Allerton Conference on Communication, Control, and Computing, Vol. 16, October 1978.
17. R. R. Mielke, R. A. Maynard, S. R. Liberty, and S. Srinathkumar: "Design of Robust State Feedback Controllers Via Eigenvalue/Eigenvector Assignment." Asilomar Conference on Circuits, Systems, and Computers, Vol. 12, November 1978.
18. S. R. Liberty, R. R. Mielke, and R. A. Maynard: "A Summary of Spectral Synthesis Procedures for Multivariable Systems." Final Report, NASA Grant NSG-1519, Old Dominion University Research Foundation, November 1979.
19. R. A. Maynard, R. R. Mielke, and S. R. Liberty: "Eigenvector Assignment for Noise Suppression." Southeastern Symposium on System Theory, Vol. 13, March 1981.
20. S. R. Liberty, R. R. Mielke, and R. A. Maynard: "Design of Multivariable Feedback Control Systems Via Spectral Assignment." Progress Report, NASA Grant NSG-1650, Old Dominion University Research Foundation, October 1980.
21. R. R. Mielke and L. J. Tung: "Design of Multivariable Feedback Control Systems Via Spectral Assignment." Progress Report, NASA Grant NSG-1650, Old Dominion University Research Foundation, October 1981.
22. T. M. Owens and R. R. Mielke: "A Spectral Assignment Gain Modification Procedure," Proceedings of Southeastern Symposium on System Theory, March 1981.

**APPENDIX**

**Eigenvalue/Eigenvector Assignment Program Listing**

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003 C-Function: Mode Selection.
00004 C-IMSL routines called: UGETIO.
00005 C-Spectral Assignment routines: MODE0 through MODE8.
00006 C-Logical devices; Input Unit: 5 Output Unit: 5
00007 C Storage Unit(s): IU=20
00008 C-Random Access Files: SYSTEM.DAT
00009 REAL A(10,10),B(10,10),C(10,10),ZERO
00010 INTEGER MODE,IDGT,NS,NI,NO
00011 COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00012 CALL UGETIO (3,5,5)
00013 C*****
00014 IRS=102
00015 IU=20
00016 OPEN (FILE='SYSTEM.DAT',ACCESS='RANDOM',RECORD SIZE=IRS
00017 1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00018 100 WRITE (5,101)
00019 101 FORMAT (1H/,1X,70(1H*),/,1X,21(1H*),
00020 129H SPECTRAL ASSIGNMENT PACKAGE ,20(1H*),//
00021 2,1X,49HENTER DESIRED MODE OF OPERATION,MODE=C,1,2,...,8:)
00022 READ (5,*) MODE
00023 IF (MODE.LE.0) GO TO 80
00024 GO TO (1,2,3,4,5,6,7,8),MODE
00025 1 CALL MODE1
00026 GO TO 99
00027 2 CALL MODE2
00028 GO TO 99
00029 3 CALL MODE3
00030 GO TO 99
00031 4 CALL MODE4
00032 GO TO 99
00033 5 CALL MODE5
00034 GO TO 99
00035 6 CALL MODE6
00036 GO TO 99
00037 7 CALL MODE7
00038 GO TO 99
00039 8 CALL MODE8
00040 GO TO 99
00041 80 CALL MODE0
00042 99 WRITE (5,102)
00043 102 FORMAT (1X,39HTERMINATE THIS RUN OR SELECT NEXT MODE:,//
00044 1,1X,18HWISH TO TERMINATE?)
00045 READ (5,*) I
00046 IF (I.LE.0) GO TO 100
00047 STOP
00048 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY.

```

00001 C*****
00002 C*****
00003 SUBROUTINE MODEL
00004 C-Function: System data entry.
00005 C-IMSL routines called: USWFM.
00006 C-Spectral Assignment routines: -
00007 C5Logical devices; Input Unit: 5 Output Unit: 5
00008 C Storage Unit(s): IU=20.
00009 C-Random Access Files: SYSTEM.DAT .
00010 REAL A(10,10),B(10,10),C(10,10),NULL(5)
00011 COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00012 IU=20
00013 210 WRITE (5,1)
00014 1 FORMAT (1X,26(1H*),19H MODE 1:DATA ENTRY ,25(1H*),//,1X,10(1H*)
00015 1,34HENTER OR CHANGE SYSTEM PARAMETERS:,//)
00016 WRITE (5,4)
00017 4 FORMAT (1X,16HPREVIOUS VALUES?)
00018 READ (5,*) I7
00019 IF (I7.GT.0) GO TO 220
00020 230 WRITE (5,2) NS,NI,NO,IDGT,ZERO
00021 2 FORMAT (5X,3HNS=,I2,10X,3HNI=,I2,10X,3HNO=,I2
00022 1,5X,5HIDGT=,I2,5X,5HZERO=,F15.12,//,1X,15HWISH TO CHANGE?)
00023 READ (5,*) I1
00024 IF (I1.LE.0) GO TO 100
00025 WRITE (5,5)
00026 5 FORMAT (1X,20HENTER NEW VALUE(S) :)
00027 READ (5,*) NS,NI,NO,IDGT,ZERO
00028 WRITE (IU*1) NS,NI,NO,IDGT,ZERO
00029 C
00030 100 CALL USWFM (10HMATRIX A :,10,A,10,NS,NS,4)
00031 WRITE (5,3)
00032 3 FORMAT (1X,15HWISH TO CHANGE?)
00033 READ (5,*) I2
00034 IF (I2.LE.0) GO TO 130
00035 WRITE (5,5)
00036 READ (5,*) ((A(I,J),J=1,NS),I=1,NS)
00037 WRITE (IU*2) ((A(I,J),J=1,NS),I=1,NS)
00038 C
00039 130 CALL USWFM (10HMATRIX B :,10,B,10,NS,NI,4)
00040 WRITE (5,3)
00041 READ (5,*) I3
00042 IF (I3.LE.0) GO TO 160
00043 WRITE (5,5)
00044 READ (5,*) ((B(I,J),J=1,NI),I=1,NS)
00045 WRITE (IU*3) ((B(I,J),J=1,NI),I=1,NS)
00046 C
00047 160 CALL USWFM (10HMATRIX C :,10,C,10,NO,NS,4)
00048 WRITE (5,3)
00049 READ (5,*) I4
00050 IF (I4.LE.0) GO TO 200
00051 WRITE (5,5)
00052 READ (5,*) ((C(I,J),J=1,NS),I=1,NO)
00053 WRITE (IU*4) ((C(I,J),J=1,NS),I=1,NO)
00054 GO TO 200
00055 C*****THIS BLOCK ACCESSED ONLY BY A GO TO 220 STATEMENT***
00056 220 CONTINUE

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00057      READ (IU'1) NS,NI,NO,IGT,ZERO
00058      READ (IU'2) ((A(I,J),J=1,NS),I=1,NS)
00059      READ (IU'3) ((B(I,J),J=1,NI),I=1,NS)
00060      READ (IU'4) ((C(I,J),J=1,NS),I=1,NO)
00061      GO TO 230
00062      C *****
00063      200 WRITE (5,6)
00064      6  FORMAT (1X,29HWISH TO EXIT FROM THIS MODE? )
00065      READ (5,*) I6
00066      IF (I6.LE.0) GO TO 210
00067      WRITE (5,7)
00068      7  FORMAT (1X,27(1H*),10H EXITING MODE 1 ,25(1H*))
00069      RETURN
00070      END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003     SUBROUTINE MQDE2
00004 C-Function: Eigenvalue Assignment.
00005 C-IMSL routines called: (USWFM).
00006 C-Spectral Assignment routines: NSA,TRANS .
00007 C-Logical devices; Input Unit: 5 Output Unit: 5
00008 C     Storage Unit(s): IU=20,IU=20+I for I=1,NS.
00009 C-Random Access Files: SYSTEM.DAT,FORxx.DAT where xx=IU=20+I for I=1,NS.
00010     REAL LRE(10),LIM(10),S(10,30),SCOPY(10,30),SP(10,10),SPP(10,20)
00011     REAL X(30,20),ML(10,10),NL(10,10)
00012     REAL NLC(10,20),PLC(10,20),MLC(10,20)
00013     REAL ALPHA(20,20),BETA(20,20),KA(20,10),KB(20,10),GAMA(20,20)
00014     REAL ACPY(20,20),AP(20,20),APP(20,10)
00015     REAL STAR(20,20),QL(10,20),RL(10,20)
00016     REAL A(10,10),B(10,10),C(10,10)
00017     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00018     COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL/EIG/LRE,LIM
00019 C***** READ SYSTEM DATA *****
00020     IRS=102
00021     IU=20
00022     READ (IU'1) NS,NI,NO,IDGT,ZERO
00023     READ (IU'2) ((A(I,J),J=1,NS),I=1,NS)
00024     READ (IU'3) ((B(I,J),J=1,NI),I=1,NS)
00025 C
00026     910     I=1
00027           WRITE (5,1)
00028           1  FORMAT (1X,20(1H*),30H MODE 2:EIGENVALUE ASSIGNMENT ,20(1H*),//
00029             1,1X,10(1H*),29H ENTER OR CHANGE EIGENVALUES://)
00030 C           WRITE (5,33) ZERO,IDGT           !***
00031 C 33     FORMAT (1X,5HZERO=,F15.12,1X,5HIDGT=,I2) !***
00032     999 CONTINUE
00033           IU=I+20
00034           IRS=202
00035           OPEN (ACCESS='RANDOM',RECORD SIZE=IRS
00036             1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00037 C
00038           WRITE (5,11)
00039           11  FORMAT (1X,16HPREVIOUS VALUES?)
00040           READ (5,*) K0
00041           IF (K0.GT.0) GO TO 12
00042           GO TO 13
00043 C
00044           12  READ (IU'1) LRE(I),LIM(I)
00045           13  WRITE (5,2) I,LRE(I),LIM(I)
00046           2  FORMAT (1X,6HLAMBDA,I2,1H:,,1X,5HREAL=,E15.6,2X,6H IMAG=
00047             1,E15.6,/,1X,15HWISH TO CHANGE?)
00048           READ (5,*) K1
00049           IF (K1.LE.0) GO TO 50
00050           write (5,14)
00051           14  format (1x,20henter new value(s) :)
00052           READ (5,*) LRE(I),LIM(I)
00053           WRITE (IU'1) LRE(I),LIM(I)
00054 C*****IS LAMBDA-I REAL OR COMPLEX?*****
00055           IF (ABS(LIM(I)).GT.ABS(ZERO)) GO TO 100
00056 C*****REAL NULL SPACE FORMULATION*****

```



ORIGINAL PAGE 19  
OF POOR QUALITY

```
00057 C*****FORM S-LAMBDA-I,(NSX(NS+NI))*****
00058 DO 10 II=1,NS
00059 DO 10 IJ=1,NS
00060 S(II,IJ)=-A(II,IJ)
00061 IF (II.EQ.IJ) S(II,IJ)=S(II,IJ)+LRE(I)
00062 10 CONTINUE
00063 INS=NS+1
00064 JNS=NS+NI
00065 DO 20 II=1,NS
00066 DO 20 IJ=INS,JNS
00067 IDUM=IJ-NS
00068 S(II,IJ)=B(II,IDUM)
00069 20 CONTINUE
00070 C CALL USWFM (11MATRIX SLI:,11,S,10,NS,JNS,4) !**
00071 C*****CALL NSA*****
00072 C WRITE (5,3) !**
00073 C 3 FORMAT (1X,'NULL SPACE OF S-LAMBDA-I,X=KL') !**
00074 C write (5,33) zero,ldgt !**
00075 CALL NSA(NS,JNS,S,10,30,X,30,20,ZERO,LDGT,SCOPY,SPP,SP)
00076 C*****PARTITION X=KL INTO NL AND ML *****
00077 DO 30 II=1,NS
00078 DO 30 IJ=1,NI
00079 NL(II,IJ)=X(II,IJ)
00080 30 CONTINUE
00081 WRITE (IU*3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
00082 DO 40 II=INS,JNS
00083 DO 40 IJ=1,NI
00084 IML=II-NS
00085 ML(IML,IJ)=X(II,IJ)
00086 40 CONTINUE
00087 WRITE (IU*4) ((ML(II,IJ),IJ=1,NI),II=1,NI)
00088 C CALL USWFM (10MATRIX NL:,:10,NL,10,NS,NI,4) !**
00089 C CALL USWFM (10MATRIX ML:,:10,ML,10,NI,NI,4) !**
00090 IF (I.GE.NS) GO TO 900
00091 50 I=I+1
00092 IF (I.GT.NS) GO TO 900
00093 WRITE (5,15)
00094 15 FORMAT (1X,16HNEXT EIGENVALUE:)
00095 GO TO 999
00096 C
00097 C
00098 100 CONTINUE
00099 C*****COMPLEX NULL SPACE FORMULATION *****
00100 C*****FORM S-LAMBDA-C, NSX(2NS+NI) *****
00101 DO 110 II=1,NS
00102 DO 110 IJ=1,NS
00103 S(II,IJ)=-A(II,IJ)
00104 IF (II.EQ.IJ) S(II,IJ)=S(II,IJ)+LRE(I)
00105 110 CONTINUE
00106 INS=NS+1
00107 NS2=2*NS
00108 NI2=2*NI
00109 DO 120 II=1,NS
00110 DO 120 IJ=INS,NS2
00111 S(II,IJ)=0.0
00112 IJDUM=IJ-NS
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00113         IF (II.EQ.IJDUM) S(II,IJ)=LIM(I)
00114     120     CONTINUE
00115         IINS=NS2+1
00116         ILC=NS2+NI
00117         DO 130 II=1,NS
00118         DO 130 IJ=IINS,ILC
00119         IJDUM=IJ-NS2
00120         S(II,IJ)=B(II,IJDUM)
00121     130     CONTINUE
00122     C       CALL USWFM (11HMATRIX SLC:,11,S,10,NS,ILC,4)   !!!
00123     C***** CALL NSA *****
00124     C       WRITE (5,4)                                     !!!
00125     C   4   FORMAT (1X,'NULL SPACE OF SLC, X=KLC ')       !!!
00126         CALL NSA (NS,ILC,S,10,30,X,30,20,ZERO,IDGT,SCOPY,SPP,SP)
00127     C***** PARTITION X=KLC INTO NLC,PLC,AND MLC *****
00128         IS=NS+NI
00129         DO 140 II=1,NS
00130         DO 140 IJ=1,IS
00131         NLC(II,IJ)=X(II,IJ)
00132     140     CONTINUE
00133         WRITE (IU'3) ((NLC(II,IJ),IJ=1,IS),II=1,NS)
00134     C
00135         DO 150 II=INS,NS2
00136         DO 150 IJ=1,IS
00137         IIDUM=II-NS
00138         PLC(IIDUM,IJ)=X(II,IJ)
00139     150     CONTINUE
00140         WRITE (IU'4) ((PLC(II,IJ),IJ=1,IS),II=1,NS)
00141     C
00142         DO 160 II=IINS,ILC
00143         DO 160 IJ=1,IS
00144         IJDUM=II-NS2
00145         MLC(IJDUM,IJ)=X(II,IJ)
00146     160     CONTINUE
00147         WRITE (IU'5) ((MLC(II,IJ),IJ=1,IS),II=1,NI)
00148     C
00149     C       CALL USWFM (11HMATRIX NLC:,11,NLC,10,NS,IS,4)   !!!
00150     C       CALL USWFM (11HMATRIX PLC:,11,PLC,10,NS,IS,4)   !!!
00151     C       CALL USWFM (11HMATRIX MLC:,11,MLC,10,NI,IS,4)   !!!
00152         IF (NS.EQ.NI) GO TO 215
00153     C***** FORM ALPHA,TRANSPOSE *****
00154         DO 170 II=1,NS2
00155         DO 170 IJ=1,IS
00156         ALPHA(II,IJ)=X(II,IJ)
00157         IF (II.GT.NS) ALPHA(II,IJ)=-X(II,IJ)
00158     170     CONTINUE
00159     C       CALL USWFM (14HMATRIX ALPHAT:,14,ALPHA,20,NS2,IS,4) !!!
00160         CALL TRANS (ALPHA,NS2,IS)
00161     C       CALL USWFM (20HTRANSPOSE OF ALPHAT:,20,ALPHA,20,IS,NS2,4) !*
00162     C***** CALL NSA *****
00163     C       WRITE (5,5)                                     !!!
00164     C   5   FORMAT (1X,'NULL SPACE OF ALPHA, KA ')         !!!
00165         CALL NSA (IS,NS2,ALPHA,20,20,KA,20,10,ZERO,IDGT,ACOPY,APP,AP)
00166         NMI=NS-NI
00167     C       CALL USWFM (10HMATRIX KA:,10,KA,20,NS2,NMI,4)   !!!
00168     C***** FORM BETA, TRANSPOSE *****

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00169      DO 180 II=1,NS
00170      DO 180 IJ=1,IS
00171      BETA(II,IJ)=PLC(II,IJ)
00172      180  CONTINUE
00173      C
00174      DO 190 II=INS,NS2
00175      DO 190 IJ=1,IS
00176      IDUM=II-NS
00177      BETA(II,IJ)=NLC(IDUM,IJ)
00178      190  CONTINUE
00179      C      CALL USWFM (13HMATRIX BETAT:,13,BETA,20,NS2,IS,4) !**
00180      CALL TRANS (BETA,NS2,IS)
00181      C      CALL USWFM (19HTRANSPOSE OF BETAT:,19,BETA,20,IS,NS2,4) !**
00182      C***** CALL NSA *****
00183      C      WRITE (5,6) !**
00184      C      6  FORMAT (1X,'NULL SPACE OF BETA, KB ') !**
00185      CALL NSA (IS,NS2,BETA,20,20,KB,20,10,ZERO,IDGT,ACOPY,APP,AP)
00186      C      CALL USWFM (10HMATRIX KB:,10,KB,20,NS2,NMI,4) !**
00187      C***** FORM GAMA, TRANSPOSE *****
00188      DO 200 II=1,NS2
00189      DO 200 IJ=1,NMI
00190      GAMA(II,IJ)=KA(II,IJ)
00191      200  CONTINUE
00192      C
00193      NMI2=2*NMI
00194      NMII=NMI+1
00195      DO 210 II=1,NS2
00196      DO 210 IJ=NMII,NMI2
00197      NMIDUM=IJ-NMI
00198      GAMA(II,IJ)=KB(II,NMIDUM)
00199      210  CONTINUE
00200      C
00201      C      CALL USWFM (13HMATRIX GAMAT:,13,GAMA,20,NS2,NMI2,4) !**
00202      CALL TRANS (GAMA,NS2,NMI2)
00203      C      CALL USWFM (19HTRANSPOSE OF GAMAT:,19,GAMA,20,NMI2,NS2,4) !**
00204      C***** CALL NSA *****
00205      C      WRITE (5,7)
00206      C      7  FORMAT (1X,'NULL SPACE OF GAMA, STAR ') !**
00207      CALL NSA (NMI2,NS2,GAMA,20,20,STAR,20,20,ZERO,IDGT,ACOPY,APP,AP)
00208      GO TO 216
00209      215  DO 216 II=1,NS2
00210      DO 216 IJ=1,NI2
00211      STAR(II,IJ)=FLOAT(0)
00212      IF (II.EQ.IJ) STAR(II,IJ)=FLOAT(1)
00213      216  CONTINUE
00214      C      CALL USWFM (12HMATRIX STAR:,12,STAR,20,NS2,NI2,4) !**
00215      C***** PARTITION STAR *****
00216      DO 220 II=1,NS
00217      DO 220 IJ=1,NI2
00218      QL(II,IJ)=STAR(II,IJ)
00219      220  CONTINUE
00220      WRITE (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
00221      C
00222      DO 230 II=INS,NS2
00223      DO 230 IJ=1,NI2
00224      IDUM=II-NS
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00225      RL(IDUM,IJ)=STAR(II,IJ)
00226      230  CONTINUE
00227      WRITE (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
00228      C      CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4)  !***
00229      C      CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4)  !***
00230      C
00231      C*****SET THE CONJUGATE VALUES *****
00232      IC=I+1
00233      IRS=202
00234      IU=IC+20
00235      OPEN (ACCESS='RANDOM',RECORD SIZE=IRS
00236      1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00237      LRE(IC)=LRE(I)
00238      LIM(IC)=-LIM(I)
00239      WRITE (IU'1) LRE(IC),LIM(IC)
00240      WRITE (5,22) IC,LRE(IC),LIM(IC)
00241      22  FORMAT (1X,6HLAMBDA,I2,6H:REAL=,E15.6,2X,6H,IMAG=,E15.6)
00242      IF (IC.GE.NS) GO TO 900
00243      I=I+2
00244      WRITE (5,15)
00245      GO TO 999
00246      C
00247      900  WRITE (5,8)
00248      8   FORMAT (1X,29HWISH TO EXIT FROM THIS MODE? )
00249      READ (5,*) KK
00250      IF (KK.LE.0) GO TO 910
00251      WRITE (5,9)
00252      9   FORMAT (1X,27(1H*),18H EXITING MODE 2 ,25(1H*))
00253      RETURN
00254      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 c*****
00002 c*****
00003       SUBROUTINE NSA(M,N,S,IIS,IJS,X,IIX,IJX,ZERO,IDGT,SCOPY,SPP,SP)
00004 C-Function: Calculates a basis for the Null Space of a MxN matrix S.
00005 C-IMSL routines called: UERSET,UERTST,LEQT2F,VSRTU,VSRTTR,(USWFM).
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: (5) Output Unit: (5)
00008 C       Storage Unit(s): -
00009 C-Random Access Files: -
00010       real s(iis,ijs),scopy(iis,ijs),spp(iis,ijs),sp(iis,ijs)
00011       real x(iix,ijx),fac,wk1(10),wk2(132),wk3(30)
00012       INTEGER M,N,IM,IN,JN,PV(30),IPV(30),K,L,DUM
00013       INTEGER PVCOPY(30),RPV(30)
00014       DO 90 I=1,M
00015       DO 90 J=1,N
00016         SCOPY(I,J)=S(I,J)
00017 90     CONTINUE
00018 C       WRITE (5,2) ZERO,IDGT                               !***
00019 C 2     FORMAT (1X,5HZERO=,F15.12,1X,5HIDGT=,I2)  !***
00020       DUM=N-M
00021       IN=N
00022       JN=N
00023       IM=1
00024       DO 20 I=1,N
00025       PV(I)=I
00026 20     CONTINUE
00027       IF (ABS(S(IM,IN)).GT.ABS(ZERO)) GO TO 30
00028 70     IN=IN-1
00029       GO TO 20
00030 30     IK=PV(JN)
00031       PV(JN)=PV(IN)
00032       IF (IN.EQ.JN) GO TO 50
00033       PV(IN)=IK
00034 C*****EXCHANGE COLUMNS IN AND JN*****
00035       DO 40 I=1,N
00036       IPV(I)=I
00037 40     CONTINUE
00038       K=IPV(IN)
00039       IPV(IN)=IPV(JN)
00040       IPV(JN)=K
00041       CALL VSRTU (S,IIS,M,N,0,IPV,WK1)
00042       IN=JN
00043 50     CONTINUE
00044       IF (IM.EQ.M) GO TO 80
00045       L=IM+1
00046 C*****GAUSSIAN PROCESS*****
00047       DO 60 IL=L,N
00048       IF (ABS(S(IL,IN)).LE.ABS(ZERO)) GO TO 60
00049       FAC=S(IL,IN)/S(IM,IN)
00050       DO 60 I=1,N
00051       S(IL,I)=S(IL,I)-FAC*S(IM,I)
00052 60     CONTINUE
00053       JN=JN-1
00054       IM=IM+1
00055       GO TO 70
00056 80     CONTINUE

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00057 C      CALL USWFM (13HS TRIANGULAR:,13,S,IIS,M,N,4)   !***
00058 C      WRITE (5,4)                                     !***
00059 C      4  FORMAT(1X , 'PERMUTATION VECTOR :')         !***
00060 C      DO 81 I=1,N                                     !***
00061 C      WRITE (5,*) PV(I)                               !***
00062 C      81  CONTINUE                                   !***
00063 C      CALL USWFM (10HMATRIX S :,10,SCOPY,IIS,M,N,4)   !***
00064 C*****SHUFFLE SCOPY,USING PVCOPY*****
00065 C      DO 120 I=1,N
00066 C      PVCOPY(I)=PV(I)
00067 C      120  CONTINUE
00068 C      CALL VSRTU (SCOPY,IIS,M,N,0,PVCOPY,WK1)
00069 C*****SCOPY NOW CONTAINS SBAR*****
00070 C      CALL USWFM (12HMATRIX SBAR:,12,SCOPY,IIS,M,N,4)   !***
00071 C*****PARTITION SBAR*****
00072 C      DO 100 I=1,M
00073 C      DO 100 J=1,DUM
00074 C      SPP(I,J)=SCOPY(I,J)
00075 C      100  CONTINUE
00076 C      CALL USWFM (11HMATRIX SPP:,11,SPP,IIS,M,DUM,4) !***
00077 C      DO 110 I=1,M
00078 C      DO 110 J=1,M
00079 C      JDUM=J+DUM
00080 C      SP(I,J)=SCOPY(I,JDUM)
00081 C      110  CONTINUE
00082 C      CALL USWFM (10HMATRIX SP:,10,SP,IIS,M,M,4)   !***
00083 C*****LINEAR EQUATION SOLUTION*****
00084 C      IT=IDGT
00085 C      CALL LEQT2F (SP,DUM,M,IIS,SPP,IT,WK2,IER)
00086 C      CALL UERSET (3,LEVOLD)
00087 C      CALL UERTST (IER,6HLEQT2F)
00088 C      WRITE (5,3) IT                                     !***
00089 C      3  FORMAT(1X ,3IHIDGT ON RETURN FROM LEQT2F IS =,I3)  !***
00090 C*****SPP CONTAINS XP*****
00091 C*****SORT PV*****
00092 C      DO 130 I=1,N
00093 C      RPV(I)=I
00094 C      130  CONTINUE
00095 C      CALL VSRTU (PV,N,RPV)
00096 C*****FORM X*****
00097 C      DO 140 I=1,DUM
00098 C      DO 140 J=1,DUM
00099 C      X(I,J)=FLOAT(0)
00100 C      IF (I.EQ.J) X(I,J)=-FLOAT(1)
00101 C      140  CONTINUE
00102 C      IIDUM=DUM+1
00103 C      DO 150 I=IIDUM,N
00104 C      DO 150 J=1,DUM
00105 C      IDUM=I-DUM
00106 C      X(I,J)=SPP(IDUM,J)
00107 C      150  CONTINUE
00108 C*****SHUFFLE ROWS OF X*****
00109 C      CALL USWFM (20HX BEFORE SHUFFLING :,20,X,IIX,N,DUM,4) !***
00110 C      CALL VSRTU (X,IIX,N,DUM,1,RPV,WK3)
00111 C      CALL USWFM (20HBASIS VECTORS ARE :,20,X,IIX,N,DUM,4) !***
00112 C      RETURN

```

00113

END

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE MODE0
00004 C-Function: Signature.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: 5
00008 C Storage Unit(s): -
00009 C-Random Access Files: -
00010 WRITE (5,1)
00011 1 FORMAT (/,13X,3H***,/,13X,3H***,16X,23HOld Dominion University, /
00012 1,13X,3H***,10X,36HDepartment of Electrical Engineering,/,13x,3H***
00013 2,21X,14HMohsen Marefat,/,13X,3H***,21X,14HSeptember 1982)
00014 WRITE (5,2)
00015 2 FORMAT (4X,3H***,3X,2H**,1X,3H***,3X,3H***,/,2X,14(1H*),3X,3H***,
00016 1/,1X,4(6H*** ),3X,31HThe Spectral Assignment Package,/,1X
00017 2,4(6H*** ),3X,31(1H=),/,2X,19(1H*),/,1X,3(6H **))
00018 WRITE (5,3)
00019 3 FORMAT (//,6X,52HDocumentation and a user guide for this CAD progr
00020 lam ,/,6x,34Hpackage is avialable upon request.,/,
00021 26X,45HContact Dr. R.R. Mielke at the EE department.)
00022 RETURN
00023 END
```

ORIGINAL PAGE 19  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE TRANS (A,IM,IN)
00004 C-Function: Returns the transpose of matrix [A] in A.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: -
00008 C Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL A(20,20),AT(20,20)
00011 DO 10 I=1,IM
00012 DO 10 J=1,IN
00013 AT(J,I)=A(I,J)
00014 10 CONTINUE
00015 DO 20 I=1,IN
00016 DO 20 J=1,IM
00017 A(I,J)=AT(I,J)
00018 20 CONTINUE
00019 RETURN
00020 END
```



ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003 SUBROUTINE MODE3
00004 C-Function: Main routine for Eigenvector Assignment.
00005 C-IMSL routines called: UERTST, UERSET, USWFV, USWFM, LLSQF, VMULFF.
00006 C-Spectral Assignment routines: GAIN, IMP, PROJ, NORM.
00007 C-Logical devices; Input Unit: 5 Output Unit: 5
00008 C Storage Unit(s): IU=20, IUT=20+NS+1, IU=20+J for J=1, NS.
00009 C-Random Access Files: SYSTEM.DAT, CURRNT.DAT, FORxx.DAT where xx=20+J
00010 C for J=1, NS.
00011 C NULL SPACE ARRAYS
00012 REAL ML(10,10), NL(10,10)
00013 REAL NLC(10,20), PLC(10,20), MLC(10,20)
00014 REAL STAR(20,20), QL(10,20), RL(10,20)
00015 C AUX. ARRAYS
00016 REAL WKAREA(130), CP(20,20), ATA(20,20), ATAI(20,20)
00017 REAL PNL(10,10), PSTAR(20,20), XX(10,10)
00018 REAL LRE(10), LIM(10)
00019 C MODE 3 ARRAYS
00020 REAL VRE(10,10), VIM(10,10), VD(20), VA(20), E(20), X(20), H(20)
00021 REAL WJ(10), W(10,10), V(10,10), VINV(10,10), F(10,10), AHAT(10,10)
00022 INTEGER IP(10)
00023 REAL A(10,10), B(10,10), C(10,10)
00024 COMMON/SYS/A, B, C, ZERO, IDGT, NS, NI, NO
00025 COMMON/AUG/F, AHAT/EIG/LRE, LIM
00026 COMMON/VEC/VA, E, X, WJ, W, XX, V, VINV
00027 COMMON/NSPA/ML, NL, NLC, PLC, MLC, STAR, QL, RL
00028 C***** READ SYSTEM DATA *****
00029 CALL UERSET (3, LEVOLD)
00030 IRS=102
00031 IU=20
00032 READ (IU'1) NS, NI, NO, IDGT, ZERO
00033 READ (IU'2) ((A(I, J), J=1, NS), I=1, NS)
00034 READ (IU'3) ((B(I, J), J=1, NI), I=1, NS)
00035 C
00036 IUT=IU+NS+1
00037 OPEN (FILE='CURRNT.DAT', ACCESS='RANDOM', RECORD SIZE=IRS
00038 1, UNIT=IUT, MODE='BINARY', DEVICE='DSK', DISPOSE='SAVE')
00039 WRITE (5, 1)
00040 1 FORMAT (1X, 20(1H*), 31H MODE 3: EIGENVECTOR ASSIGNMENT , 19(1H*)
00041 1, //, 1X, 10(1H*), 30H ENTER OR CHANGE EIGENVECTORS: , //)
00042 WRITE (5, 11)
00043 11 FORMAT (1X, 16HPREVIOUS VALUES?)
00044 READ (5, *) KO
00045 IF (KO.LE.0) GO TO 910
00046 IFLAG=1
00047 READ (IUT'1) ((V(II, IJ), IJ=1, NS), II=1, NS)
00048 READ (IUT'2) ((XX(II, IJ), IJ=1, NS), II=1, NI)
00049 910 J=1
00050 999 CONTINUE
00051 IU=J+20
00052 IRS=202
00053 OPEN (ACCESS='RANDOM', RECORD SIZE=IRS, UNIT=IU
00054 1, MODE='BINARY', DEVICE='DSK', DISPOSE='SAVE')
00055 IF (IFLAG.NE.1) GO TO 13
00056 READ (IU'2) ((VRE(IV, J), VIM(IV, J)), IV=1, NS)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00057      13      WRITE (5,14) J
00058      14      FORMAT (1X,13HEIGENVECTOR V,I2,1H:,3X,6H(REAL),14X,6H(IMAG))
00059          DO 10 IV=1,NS
00060          WRITE (5,15) VRE(IV,J),VIM(IV,J)
00061      15      FORMAT (15X,E15.6,5X,E15.6)
00062      10      CONTINUE
00063          WRITE (5,16)
00064      16      FORMAT (1X,15HWISH TO CHANGE?)
00065          READ (5,*) K1
00066          IF (K1.LE.0) GO TO 50
00067          WRITE (5,17)
00068      17      FORMAT (1X,28HENTER A NEW DESIRED VECTOR :)
00069          READ (5,*) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
00070          WRITE (IU'2) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
00071      C***** IS V-J REAL OR COMPLEX ? *****
00072          READ (IU'1) LRE(J),LIM(J)
00073          IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 100
00074      C***** REAL EIGENVECTOR PROJECTION *****
00075          READ (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
00076      C          CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4)   !**
00077          CALL PRDJ (NL,NS,NI,10,10,PNL,CP,ATA,ATAI,IDGT)
00078      C          CALL USWFM (11HMATRIX PNL:,11,PNL,10,NS,NS,4)   !**
00079      C***** PROJECT VD ONTO COLUMN SPACE OF N-LAMBDA **
00080          DO 30 IV=1,NS
00081          VD(IV)=VRE(IV,J)
00082      30      CONTINUE
00083          CALL USWFV(15HDESIRED VECTOR:.,15,VD,NS,1,4)
00084          CALL VMULFF (PNL,VD,NS,NS,1,10,20,VA,20,IER)
00085      C          CALL UERTST (IER,6HVMULFF)
00086          CALL USWFV (15HACTUAL VECTORT:.,15,VA,NS,1,4)
00087      C          CALL USWFM (15HYA FROM USWFM :.,15,VA,20,NS,1,4)**
00088      C***** FIND THE ERROR VECTOR *****
00089          CALL IMP(PNL,NS,10)
00090      C          CALL USWFM (13HMATRIX I-PNL:.,13,PNL,10,NS,NS,4)**
00091          CALL VMULFF (PNL,VD,NS,NS,1,10,20,E,20,IER)
00092      C          CALL UERTST (IER,6HVMULFF)
00093          CALL USWFV (14HERROR VECTORT:.,14,E,NS,1,4)
00094          CALL NORM (VD,NS,XVD)
00095          CALL NORM (VA,NS,XVA)
00096          CALL NORM (E,NS,XE)
00097          WRITE (5,18) XVD,XVA,XE
00098      18      FORMAT (1X,31HLENGTH OF THE DESIRED VECTOR =,F15.6,/
00099          1,1X,31HLENGTH OF THE PROJECTED VECTOR=,F15.6,/
00100          2,1X,31HLENGTH OF THE ERROR VECTOR =,F15.6)
00101          WRITE (5,21)
00102      21      FORMAT (1X,24HIS THE ERROR ACCEPTABLE?)
00103          READ (5,*) KK
00104          IF (KK.GT.0) GO TO 45
00105          WRITE (5,17)
00106          READ (5,*) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
00107          WRITE (IU'2) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
00108          CALL IMP(PNL,NS,10)
00109          GO TO 25
00110      45      DO 46 IV=1,NS
00111          V(IV,J)=VA(IV)
00112      46      CONTINUE

```

```

00113 C***** SOLVE NL*X=VA FOR X *****
00114 C NOTE: VA IS DESTROYED!
00115 CALL LLSQF (NL,10,NS,NI,VA,-1.0,NI,X,H,IP,IER)
00116 CALL UERTST (IER,6HLLSQF )
00117 C CALL USWFV (10HVECTOR XT:,10,X,NI,1,4) !!
00118 DO 49 IV=1,NI
00119 XX(IV,J)=X(IV)
00120 49 CONTINUE
00121 IF (J.GE.NS) GO TO 900
00122 50 J=J+1
00123 IF (J.GT.NS) GO TO 900
00124 WRITE (5,19)
00125 19 FORMAT (1X,17HNEXT EIGENVECTOR:)
00126 GO TO 999
00127 C
00128 100 CONTINUE
00129 C***** COMPLEX EIGENVECTOR ASSIGNMENT *****
00130 IS=NS+NI
00131 NI2=2*NI
00132 NS2=2*NS
00133 INS=NS+1
00134 READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
00135 READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
00136 C CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4) !!
00137 C CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4) !!
00138 C***** FORM STAR AND FIND P-STAR *****
00139 DO 105 II=1,NS
00140 DO 105 IJ=1,NI2
00141 STAR(II,IJ)=QL(II,IJ)
00142 105 CONTINUE
00143 DO 110 II=INS,NS2
00144 DO 110 IJ=1,NI2
00145 IDUM=II-NS
00146 STAR(II,IJ)=RL(IDUM,IJ)
00147 110 CONTINUE
00148 CALL PROJ (STAR,NS2,NI2,20,20,PSTAR,CP,ATA,ATAI,IDGT)
00149 C CALL USWFM (12HMATRIX STAR:,12,PSTAR,20,NS2,NI2,4) !!
00150 C CALL USWFM (13HMATRIX PSTAR:,13,PSTAR,20,NS2,NS2,4) !!!
00151 C***** PROJECT VD CNTO THE COLUMN SPACE OF STAR *****
00152 114 DO 115 IV=1,NS
00153 VD(IV)=VRE(IV,J)
00154 115 CONTINUE
00155 DO 120 IV=INS,NS2
00156 IVDUM=IV-NS
00157 VD(IV)=VIM(IVDUM,J)
00158 120 CONTINUE
00159 CALL USWFV (11HCOMPLEX VD:,11,VD,NS2,1,4)
00160 CALL VMULFF (PSTAR,VD,NS2,NS2,1,20,20,VA,20,IER)
00161 C CALL UERTST (IER,6HVMULFF)
00162 CALL USWFV (15HACTUAL VECTORT:,15,VA,NS2,1,4)
00163 C***** FIND THE ERROR VECTOR *****
00164 CALL IMP(PSTAR,NS2,20)
00165 C CALL USWFM (15HMATRIX I-PSTAR:,15,PSTAR,20,NS2,NS2,4) !!
00166 CALL VMULFF (PSTAR,VD,NS2,NS2,1,20,20,E,20,IER)
00167 C CALL UERTST (IER,6HVMULFF)
00168 CALL USWFV (14HERROR VECTORT:,14,E,NS2,1,4)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00169      CALL NORM (VD,NS2,XVD)
00170      CALL NORM (VA,NS2,XVA)
00171      CALL NGRM (E,NS2,XE)
00172      WRITE (5,18) XVD,XVA,XE
00173      WRITE (5,21)
00174      READ (5,*) KM
00175      IF (KM.GT.0) GO TO 134
00176      WRITE (5,17)
00177      READ (5,*) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
00178      WRITE (IU*2) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
00179      CALL IMP(PSTAR,NS2,20)
00180      GO TO 114
00181 134     IC=J+1
00182      DO 136 IV=1,NS
00183      V(IV,J)=VA(IV)
00184      IVNS=IV+NS
00185      V(IV,IC)=VA(IVNS)
00186 136     CONTINUE
00187      CALL LLSQF (STAR,20,NS2,NI2,VA,-1.0,NI2,X,H,IP,IER)
00188      CALL UERTST (IER,6HLLSQF )
00189 C      CALL USWFV (16H[XX]-J,[XX]-J+1:,16,X,NI2,1,4)   !***
00190      DO 137 IV=1,NI
00191      XX(IV,J)=X(IV)
00192      IVNS=IV+NI
00193      XX(IV,IC)=X(IVNS)
00194 137     CONTINUE
00195 C      CALL USWFM (13HMATRIX [XX]:,13,XX,10,NI,NS,4)   !***
00196 C***** SET THE CONJUGATE VALUES *****
00197      IRS=202
00198      IU=IC+20
00199      OPEN (ACCESS='RANDOM',RECORD SIZE=IRS
00200      1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00201      DO 220 IV=1,NS
00202      VRE(IV,IC)=VRE(IV,J)
00203      VIM(IV,IC)=-VIM(IV,J)
00204 220     CONTINUE
00205      WRITE (IU*2) ((VRE(IV,IC),VIM(IV,IC)),IV=1,NS)
00206      WRITE (5,14) IC
00207      DO 230 IV=1,NS
00208      WRITE (5,15) VRE(IV,IC),VIM(IV,IC)
00209 230     CONTINUE
00210      IF (IC.GE.NS) GO TO 900
00211      J=J+1
00212      GO TO 50
00213 900     CONTINUE
00214      WRITE (5,901)
00215 901     FORMAT(1X,49H=====CONTENTS OF "CURRNT" DATA FILE INCLUDE:)
00216 C      CALL USWFM (13HMATRIX [XX]:,13,XX,10,NI,NS,4)   !***
00217      CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)   !***
00218      WRITE (5,902)
00219 902     FORMAT(1X,44HWISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
00220      READ (5,*) KS
00221      IF (KS.LE.0) GO TO 903
00222      CALL DSPLAY(NS,ZERO)
00223 903     CALL GAIN
00224      CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)   !***

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00225      CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)      !***
00226      WRITE (5,8)
00227      8      FORMAT (1X,29HWISH TO EXIT FROM THIS MODE? )
00228      READ (5,*) KT
00229      IF (KT.GT.0) GO TO 920
00230      IFLAG=1
00231      GO TO 910
00232      920     WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00233      WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00234      WRITE (IUT'3) ((W(II,IJ),IJ=1,NS),II=1,NI)
00235      WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00236      WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00237      WRITE (5,9)
00238      9      FORMAT (1X,27(1H*),18H  EXITING MODE 3  ,25(1H*))
00239      RETURN
00240      END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 c*****
00002 c*****
00003 SUBROUTINE PROJ(A,M,N,IM,IN,P,CP,ATA,ATAI,IDGT)
00004 C-Function: Calculates a projection matrix [P] for the allowable
00005 C- space represented by [A].
00006 C-IMSL routines called: UERSET,UERTST,LINV2F,VMULFF,VMULFM,VMULFP,
00007 C (USWFM).
00008 C-Spectral Assignment routines: -
00009 C-Logical devices; Input Unit: - Output Unit: (5)
00010 C Storage Unit(s): -
00011 C-Random Access Files: -
00012 REAL A(IM,IN),ATA(N,N),ATAI(N,N)
00013 REAL P(IM,IM),CP(N,M),WKAREA(460)
00014 CALL UERSET (3,LEVOLD)
00015 C CALL USWFM(9HMATRIX A:,9,A,IM,M,N,4) !**
00016 DO 10 I=1,N
00017 DO 10 J=1,N
00018 ATAI(I,J)=FLOAT(0)
00019 IF (I.EQ.J) ATAI(I,J)=FLOAT(1)
00020 10 CONTINUE
00021 CALL VMULFM (A,A,M,N,N,IM,IM,ATA,N,IER)
00022 C CALL UERTST (IER,6HVMULFM)
00023 C CALL USWFM (11HMATRIX ATA:,11,ATA,N,N,N,4) !**
00024 CALL LINV2F (ATA,N,N,ATAI,IDGT,WKAREA,IER)
00025 CALL UERTST (IER,6HLINV2F)
00026 C CALL USWFM (12HMATRIX ATAI:,12,ATAI,N,N,N,4) !**
00027 CALL VMULFP (ATAI,A,N,N,M,N,IM,CP,N,IER)
00028 CALL UERTST (IER,6HVMULFP)
00029 C CALL USWFM (10HMATRIX CP:,10,CP,N,N,M,4) !**
00030 CALL VMULFF (A,CP,M,N,M,IM,N,P,IM,IER)
00031 CALL UERTST (IER,6HVMULFF)
00032 C CALL USWFM (10HMATRIX P :,10,P,IM,M,M,4) !**
00033 RETURN
00034 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE NORM(V,N,XNORM)
00004 C-Function: Calculates the norm of an N-vector V.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: -
00008 C Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL V(N)
00011 XNORM=FLOAT(0)
00012 DO 10 I=1,N
00013 XNORM=XNORM+V(I)**2
00014 10 CONTINUE
00015 XNORM=SQRT(XNORM)
00016 RETURN
00017 END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003     SUBROUTINE IMP(P,N,IN)
00004 C-Function: Returns [P]=[I]-[P].
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit:  -   Output Unit:  -
00008 C           Storage Unit(s): -
00009 C-Random Access Files: -
00010     REAL P(IN,IN)
00011     DO 10 I=1,N
00012     DO 10 J=1,N
00013     P(I,J)=-P(I,J)
00014     IF (I.EQ.J) P(I,J)=P(I,J)+FLOAT(1)
00015 10    CONTINUE
00016     return
00017     end
```



ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003     SUBROUTINE GAIN
00004 C-Function: Calculates the Gain matrix,[F].
00005 C-IMSL routines called: UERSET,UERTST,LINV2F,LLSQF,VMULFF,(USWFM,USWFV).
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: (5)
00008 C           Storage Unit(s): IU=20+J for J=1,NS.
00009 C-Random Access Files: FORxx.DAT where xx=20+J for J=1,NS.
00010 C NULL SPACE ARRAYS
00011     REAL ML(10,10),NL(10,10)
00012     REAL NLC(10,20),PLC(10,20),MLC(10,20)
00013     REAL STAR(20,20),QL(10,20),RL(10,20)
00014 C AUX. ARRAYS
00015     REAL WKAREA(130),H(20)
00016 C MODE 3 ARRAYS
00017     REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00018     REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00019     INTEGER IP(10)
00020     REAL A(10,10),B(10,10),C(10,10)
00021     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00022     COMMON/AUG/F,AHAT/EIG/LRE,LIM
00023     COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00024     COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00025     CALL UERSET (3,LEVOLD)
00026 C           WRITE (5,1)
00027 C 1     FORMAT (1X,'SUBROUTINE GAIN ++++++')
00028     IRS=202
00029     J=1
00030     10   IU=J+20
00031     OPEN (ACCESS='RANDOM',RECORD SIZE=IRS,UNIT=IU
00032           1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00033 C***** Is Lambda-J real? *****
00034     READ (IU'1) LRE(J),LIM(J)
00035     IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 30
00036 C***** Find real WJ=J-th column of [W] *****
00037     READ (IU'4) ((ML(II,IJ),IJ=1,NI),II=1,NI)
00038 C           CALL USWFM (10HMATRIX ML:,10,ML,10,NI,NI,4)   !***
00039     DO 20 IV=1,NI
00040     X(IV)=XX(IV,J)
00041     20   CONTINUE
00042 C***** FORM WJ=[ML]*X AND PUT WJ IN J-TH COLUMN OF [W] ***
00043     CALL VMULFF (ML,X,NI,NI,1,10,20,WJ,10,IER)
00044     CALL UERTST (IER,6HVMULFF)
00045 C           CALL USWFV (10HVECTOR WJ:,10,WJ,NI,1,4)   !***
00046     DO 25 IV=1,NI
00047     W(IV,J)=WJ(IV)
00048     25   CONTINUE
00049     29   IF (J.GE.NS) GO TO 100
00050     J=J+1
00051     GO TO 10
00052 C***** Find complex WJ's *****
00053     30   IS=NS+NI
00054     NI2=2*NI
00055     NS2=2*NS
00056     INS=NS+1

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00057 READ (IU'3) ((NLC(II,IJ),IJ=1,IS),II=1,NS)
00058 READ (IU'4) ((PLC(II,IJ),IJ=1,IS),II=1,NS)
00059 READ (IU'5) ((MLC(II,IJ),IJ=1,IS),II=1,NI)
00060 C CALL USWFM (11HMATRIX NLC:,11,NLC,10,NS,IS,4) !!!
00061 C CALL USWFM (11HMATRIX PLC:,11,PLC,10,NS,IS,4) !!!
00062 C CALL USWFM (11HMATRIX MLC:,11,MLC,10,NI,IS,4) !!!
00063 IC=J+1
00064 C***** FORM ALPHAT AND SOLVE [ALPHAT]*X=VA FOR X *****
00065 DO 135 II=1,NS
00066 DO 135 IJ=1,IS
00067 STAR(II,IJ)=NLC(II,IJ)
00068 135 CONTINUE
00069 DO 140 II=INS,NS2
00070 DO 140 IJ=1,IS
00071 IDUM=II-NS
00072 STAR(II,IJ)=-PLC(IDUM,IJ)
00073 140 CONTINUE
00074 C CALL USWFM (14HMATRIX ALPHAT:,14,STAR,20,NS2,IS,4) !!!
00075 DO 40 IV=1,NS
00076 VA(IV)=V(IV,J)
00077 E(IV)=VA(IV)
00078 40 CONTINUE
00079 DO 50 IV=INS,NS2
00080 IVDUM=IV-NS
00081 VA(IV)=V(IVDUM,IC)
00082 E(IV)=VA(IV)
00083 50 CONTINUE
00084 CALL LLSQF (STAR,20,NS2,IS,VA,-1.0,IS,X,H,IP,IER)
00085 CALL UERTST (IER,6HLLSQF )
00086 C CALL USWFV (10HVECTOR XT:,10,X,IS,1,4) !!!
00087 C***** FORM WJ=[MLC]*XC AND PUT WJ IN THE J-TH COLUMN OF [W] *
00088 CALL VMULFF (MLC,X,NI,IS,1,10,20,WJ,10,IER)
00089 CALL UERTST (IER,6HVMULFF)
00090 C CALL USWFV (10HVECTOR WJ:,10,WJ,NI,1,4) !!!
00091 DO 60 IV=1,NI
00092 W(IV,J)=WJ(IV)
00093 60 CONTINUE
00094 C
00095 IF (J.EQ.IC) GO TO 29
00096 J=IC
00097 C***** FORM BETAT AND SOLVE [BETAT]*X=E(=VA) FOR X *****
00098 DO 180 IV=1,IS
00099 X(IV)=FLOAT(0)
00100 180 CONTINUE
00101 DO 185 II=1,NS
00102 DO 185 IJ=1,IS
00103 STAR(II,IJ)=PLC(II,IJ)
00104 185 CONTINUE
00105 DO 190 II=INS,NS2
00106 DO 190 IJ=1,IS
00107 IDUM=II-NS
00108 STAR(II,IJ)=NLC(IDUM,IJ)
00109 190 CONTINUE
00110 C CALL USWFM (13HMATRIX BETAT:,13,STAR,20,NS2,IS,4) !!!
00111 DO 70 IV=1,NS2
00112 VA(IV)=E(IV)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00113      70      CONTINUE
00114      GO TO 50
00115      C***** Print [W],-[W], then find [F], and [AHAT] *****
00116      100      CONTINUE
00117      C          CALL USWFM (11HMATRIX [W]:,11,W,10,NI,NS,4)          !***
00118      DO 80 II=1,NI
00119      DO 80 IJ=1,NS
00120      W(II,IJ)=-W(II,IJ)
00121      80      CONTINUE
00122      C          CALL USWFM (12HMATRIX -[W]:,12,W,10,NI,NS,4)          !***
00123      C          CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)          !***
00124      CALL LINV2F (V,NS,10,VINV,IDGT,WKAREA,IER)
00125      CALL UERTST (IER,6HLINV2F)
00126      C          CALL USWFM (12HMATRIX VINV:,12,VINV,10,NS,NS,4)          !***
00127      CALL VMULFF (W,VINV,NI,NS,NS,10,10,F,10,IER)
00128      CALL UERTST (IER,6HVMULFF)
00129      C          CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)          !***
00130      CALL VMULFF (B,F,NS,NI,NS,10,10,AHAT,10,IER)
00131      CALL UERTST (IER,6HVMULFF)
00132      C          CALL USWFM (4HB*F:,4,AHAT,10,NS,NS,4)          !***
00133      DO 240 II=1,NS
00134      DO 240 IJ=1,NS
00135      AHAT(II,IJ)=AHAT(II,IJ)+A(II,IJ)
00136      240      CONTINUE
00137      C          CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)          !***
00138      C          WRITE (5,2)
00139      C          2      FORMAT (1X,'EXITING SUBROUTINE GAIN -----')
00140      RETURN
00141      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE MODE8
00004 C-Function: Facilitates storage and handling of CURRENT data.
00005 C-IMSL routines called: UERSET.
00006 C-Spectral Assignment routines: -.
00007 C-Logical devices; Input Unit: 5 Output Unit: 5
00008 C Storage Unit(s): IU=20,IUT=20+ns+1,IBAK=IUT+I for I=1,9.
00009 C-Random Access Files: SYSTEM.DAT,CURRNT.DAT,FQRxx.DAT where xx=IBAK.
00010 REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
00011 REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
00012 REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00013 REAL A(10,10),B(10,10),C(10,10)
00014 COMMON/SYS/A,B,C,ZERO, IDGT,NS,NI,NO
00015 COMMON/AUG/F,AHAT/AUX/AUX1,AUX2,AUX3
00016 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00017 CALL UERSET(3,LEVOLD)
00018 IU=20
00019 READ (IU'1) NS,NI,NO, IDGT,ZERO
00020 WRITE (5,11)
00021 11 FORMAT (1X,23(1H*),22H MODE 8:DATA TRANSFER ,25(1H*),//,
00022 11X,54HENTER # OF BACKUP FILE YOU WISH TO ADDRESS,I=1,...,9 :)
00023 READ (5,*) I
00024 WRITE (5,12)
00025 12 FORMAT (1X,48HSET TRANSFER OPTIONS:--1 FOR [CURRNT]==>[BAKUPI],/
00026 1,22X,27H--2 FOR [CURRNT]<==[BAKUPI],/
00027 2,22X,28H--3 FOR [CURRNT]<==>[BAKUPI])
00028 READ (5,*) IOP
00029 IUT=20+NS+1
00030 OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=102
00031 1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00032 IBAK=IUT+I
00033 OPEN (ACCESS='RANDOM',RECORD SIZE=102
00034 1,UNIT=IBAK,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00035 IF (IOP.EQ.2) GO TO 20
00036 READ (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00037 READ (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00038 READ (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00039 READ (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00040 C CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4) !**
00041 C CALL USWFM (10HMATRIX XX :,10,XX,10,NI,NS,4) !**
00042 C CALL USWFM (14HGAIN MATRIX F :,14,F,10,NI,NS,4) !**
00043 C CALL USWFM (12HMATRIX AHAT :,12,AHAT,10,NS,NS,4) !**
00044 IF (IOP.EQ.3) GO TO 30
00045 WRITE (IBAK'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00046 WRITE (IBAK'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00047 WRITE (IBAK'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00048 WRITE (IBAK'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00049 GO TO 999
00050 30 DO 34 II=1,NS
00051 DO 34 IJ=1,NS
00052 AUX1(II,IJ)=V(II,IJ)
00053 AUX2(II,IJ)=AHAT(II,IJ)
00054 34 CONTINUE
00055 READ (IBAK'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00056 READ (IBAK'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00057 WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00058 WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00059      WRITE (IBAK'1) ((AUX1(II,IJ),IJ=1,NS),II=1,NS)
00060      WRITE (IBAK'5) ((AUX2(II,IJ),IJ=1,NS),II=1,NS)
00061      DO 35 II=1,NI
00062      DO 35 IJ=1,NS
00063      AUX2(II,IJ)=XX(II,IJ)
00064      AUX3(II,IJ)=F(II,IJ)
00065      35 CONTINUE
00066      READ (IBAK'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00067      READ (IBAK'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00068      WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00069      WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=2,NI)
00070      WRITE (IBAK'2) ((AUX2(II,IJ),IJ=1,NS),II=1,NI)
00071      WRITE (IBAK'4) ((AUX3(II,IJ),IJ=1,NS),II=1,NI)
00072      GO TO 999
00073      20 READ (IBAK'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00074      READ (IBAK'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00075      READ (IBAK'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00076      READ (IBAK'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00077      WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00078      WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00079      WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00080      WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00081      999 GO TO (1,2,3),IOP
00082      1  WRITE (5,13) I
00083      13  FORMAT (10X,17H[CURRNT]==>[BAKUP,I1,1H])
00084      GO TO 900
00085      2  WRITE (5,14) I
00086      14  FORMAT (10X,17H[CURRNT]<=>[BAKUP,I1,1H])
00087      GO TO 900
00088      3  WRITE (5,15) I
00089      15  FORMAT (10X,18H[CURRNT]<=>[BAKUP,I1,1H])
00090      900 RETURN
00091      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003 SUBROUTINE DSPLAY(NS,ZERO)
00004 C-Function: Displays normalized Eigenvectos.
00005 C-IMSL routines called: USWFM.
00006 C-Spectral Assignment routines: NORM.
00007 C-Logical devices; Input Unit: - Output Unit: 5
00008 C Storage Unit(s): IU=20+J for J=1,NS.
00009 C-Random Access Files: FORxx.DAT where xx=20+J for J=1,NS.
00010 REAL MAT(10,10),LRE(10),LIM(10),VA(20),E(20),X(20),WJ(10)
00011 REAL W(10,10),XX(10,10),V(10,10),VINV(10,10)
00012 COMMON/VEC/VA,E,X,W,XX,V,VINV/EIG/LRE,LIM
00013 J=1
00014 10 IU=J+20
00015 READ (IU,1) LRE(J),LIM(J)
00016 IF (ABS(LIM(J)).GT.ZERO) GO TO 100
00017 DO 20 I=1,NS
00018 VA(I)=V(I,J)
00019 20 CONTINUE
00020 CALL NORM(VA,NS,XVA)
00021 DO 30 I=1,NS
00022 MAT(I,J)=VA(I)/XVA
00023 30 CONTINUE
00024 GO TO 200
00025 100 NS2=2*NS
00026 JC=J+1
00027 DO 120 I=1,NS2
00028 IF (I.GT.NS) GO TO 110
00029 VA(I)=V(I,J)
00030 GO TO 120
00031 110 INS=I-NS
00032 VA(I)=V(INS,JC)
00033 120 CONTINUE
00034 CALL NORM(VA,NS2,XVA)
00035 DO 140 I=1,NS2
00036 IF (I.GT.NS) GO TO 130
00037 MAT(I,J)=VA(I)/XVA
00038 GO TO 140
00039 130 INS=I-NS
00040 MAT(INS,JC)=VA(I)/XVA
00041 140 CONTINUE
00042 J=J+1
00043 200 IF(J.GE.NS) GO TO 300
00044 J=J+1
00045 GO TO 10
00046 300 CALL USWFM(20HNORMALIZED VECTORS : ,20,MAT,10,NS,NS,4)
00047 RETURN
00048 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003     SUBROUTINE MODE4
00004 C-Function: Simulates and plots time responses.
00005 C-IMSL routines called: UERSET,UERTST,UGETIO,DVERK,VMULFF,USPLO.
00006 C-Spectral Assignment routines: UEVAL,FCN.
00007 C-Logical devices; Input Unit: 5 Output Unit: 5
00008 C     Storage Unit(s): IU=20,IUT=20+NS+1.
00009 C-Random Access Files: SYSTEM.DAT,CURRNT.DAT .
00010     INTEGER INOPT(10)
00011     REAL AMP(10),SLOPE(10),U(10),ATIL(10,10),CCNST(10)
00012     REAL X(10),CX(24),W(10,10),Y(10),XPRIME(10)
00013 C PLOT ARRAYES
00014     REAL T(201),UMAT(201,10),XMAT(201,10),VEC(201),RANGE(4)
00015     REAL IMAG4(5151)
00016 C SYSTEM ARRAYES
00017     REAL A(10,10),B(10,10),C(10,10),F(10,10),AHAT(10,10)
00018     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00019     COMMON/AUG/F,AHAT
00020     COMMON/DIF/ATIL,CONST
00021     EXTERNAL FCN
00022     CALL UERSET(3,LEVOLD)
00023 C***** READ SYSTEM DATA *****
00024     IRS=102
00025     IU=20
00026     READ (IU'1) NS,NI,NO,IDGT,ZERO
00027     READ (IU'2) ((A(I,J),J=1,NS),I=1,NS)
00028     READ (IU'3) ((B(I,J),J=1,NI),I=1,NS)
00029     READ (IU'4) ((C(I,J),J=1,NS),I=1,NO)
00030     IUT=IU+NS+1
00031     OPEN(FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=IRS
00032     ,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00033     READ (IUT'4) ((F(I,J),J=1,NS),I=1,NI)
00034     READ (IUT'5) ((AHAT(I,J),J=1,NS),I=1,NS)
00035 C
00036 180 WRITE (5,11)
00037 11 FORMAT (1X,23(1H*),24H MODE 4:TIME SIMULATION ,23(1H*),//,1X,10(1H
00038 1*),27H CHOOSE SIMULATION OPTIONS: ,/,1X,1H-,62HENTER: 1 TO SIMULATE
00039 2 [A], 2 TO SIMULATE [AHAT],(3 FOR [ATIL]):)
00040 177 CONTINUE
00041     DO 178 II=1,201
00042     T(II)=FLOAT(0)
00043     VEC(II)=FLOAT(0)
00044 178 CONTINUE
00045     READ (5,*) ISYS
00046     GO TO (1,2,3),ISYS
00047 1     DO 10 I=1,NS
00048     DO 10 J=1,NS
00049     ATIL(I,J)=A(I,J)
00050 10 CONTINUE
00051     GO TO 30
00052 2     DO 20 I=1,NS
00053     DO 20 J=1,NS
00054     ATIL(I,J)=AHAT(I,J)
00055 20 CONTINUE
00056     GO TO 30

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00057      3      WRITE (5,12)
00058      12      FORMAT (1X,36HENTER SYSTEM MATRIX TO BE SIMULATED:)
00059      READ (5,*) ((ATIL(I,J),J=1,NS),I=1,NS)
00060      30      CONTINUE
00061      WRITE (5,13)
00062      13      FORMAT (1X,58HENTER 0 TO SIMULATE OUTPUTS,1 TO SIMULATE STATE VARI
00063      TABLES:)
00064      READ (5,*) IOUT
00065      WRITE (5,14)
00066      14      FORMAT (1X,47HENTER SIMULATION TIME,(REAL NUMBER IN SECONDS:))
00067      READ (5,*) DT
00068      WRITE (5,15)
00069      15      FORMAT (1X,50HENTER NUMBER OF POINTS TO BE CALCULATED,(200 MAX:))
00070      READ (5,*) NP
00071      WRITE (5,16)
00072      16      FORMAT (1X,31HSPECIFY THE INITIAL CONDITIONS:)
00073      DO 40 I=1,NS
00074      WRITE (5,17) I
00075      17      FORMAT (1X,1HX,I2,4H(0:))
00076      READ (5,*) X(I)
00077      40      CONTINUE
00078      WRITE (5,32)
00079      32      FORMAT (1X,56HCHOOSE INPUT OPTIONS:1 FOR NO INPUT, 2 FOR A STEP IN
00080      IPUT,/,1X,21(1H ),40H3 FOR A RAMP,AND 4 FOR A TRUNCATED RAMP:)
00081      DO 50 I=1,NI
00082      WRITE (5,18) I
00083      18      FORMAT (1X,18HINPUT OPTION FOR U,I2,1H:)
00084      READ (5,*) INOPT(I)
00085      IF (INOPT(I).NE.2) GO TO 51
00086      WRITE (5,19) I
00087      19      FORMAT (1X,37HSPECIFY AMPLITUDE OF THE STEP INPUT U,I2,1H:)
00088      READ (5,*) AMP(I)
00089      GO TO 50
00090      51      IF (INOPT(I).NE.3) GO TO 52
00091      WRITE (5,21) I
00092      21      FORMAT (1X,33HSPECIFY SLOPE OF THE RAMP INPUT U,I2,1H:)
00093      READ (5,*) SLOPE(I)
00094      GO TO 50
00095      52      IF (INOPT(I).NE.4) GO TO 50
00096      WRITE (5,22) I
00097      22      FORMAT (1X,33HSPECIFY AMPLITUDE AND SLOPE FOR U,I2,1H:)
00098      READ (5,*) AMP(I),SLOPE(I)
00099      50      CONTINUE
00100      C***** DIFFERENTIAL EQUATION SOLUTION *****
00101      IND=1
00102      TOL=ZERO*100.000000
00103      TINT=DT/NP
00104      NPI=NP+1
00105      DO 100 K=1,NPI
00106      KMI=K-1
00107      TEND=FLOAT(KMI)*TINT
00108      CALL UEVAL (INOPT,AMP,SLOPE,U,NI,TEND)
00109      CALL VMULFF (3,U,NS,NI,1,10,10,CONST,10,IER)
00110      CALL UERTST (IER,6HVMULFF)      !***
00111      CALL OVERK (NS,FCN,T,X,TEND,TOL,IND,CX,10,W,IER)
00112      IF (IND.LT.0.OR.IER.GT.0) GO TO 190

```



```

00113      53      T(K)=TEND
00114      DO 60 J=1,NI
00115      UMAT(K,J)=U(J)
00116      60      CONTINUE
00117      IF (IOUT.EQ.0) GO TO 80
00118      DO 70 J=1,NS
00119      XMAT(K,J)=X(J)
00120      70      CONTINUE
00121      N=NS
00122      GO TO 100
00123      80      CALL VMULFF (C,X,NO,NS,1,10,10,Y,10,IER)
00124      C      CALL UERTST (IER,6HVMULFF)          !***
00125      DO 90 J=1,NO
00126      XMAT(K,J)=Y(J)
00127      90      CONTINUE
00128      N=NO
00129      100     CONTINUE
00130      C***** PLOT *****
00131      WRITE (5,23)
00132      23      FORMAT (1X,49HENTER 0 FOR 80 DISPLAY COLUMNS,1 FOR 129 COLUMNS:)
00133      READ (5,*) IOPT
00134      115     WRITE (5,24)
00135      24      FORMAT (1X,48HENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:)
00136      READ (5,*) IPLOT
00137      WRITE (5,25)
00138      25      FORMAT (1X,51HDO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?)
00139      READ (5,*) IRANGE
00140      IF (IRANGE.GT.0) GO TO 120
00141      DO 110 I=1,4
00142      RANGE(I)=0.0
00143      110     CONTINUE
00144      GO TO 124
00145      120     WRITE (5,26)
00146      26      FORMAT (1X,41HENTER MIN X,MAX X,MIN Y,AND MAX Y VALUES:)
00147      READ (5,*) (RANGE(I),I=1,4)
00148      C***** PLOT INPUTS *****
00149      124     DO 125 J=1,NI
00150      IF (INOPT(J).NE.1) GO TO 130
00151      125     CONTINUE
00152      GO TO 135
00153      130     CONTINUE
00154      WRITE (5,33)
00155      33      FORMAT (1X,50HPOSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER,/
00156      1,1X,41HYOU MAY ADD A SHORT NOTE (20 CHARACTERS.))
00157      READ (5,34) III
00158      34      FORMAT (I1,20X)
00159      CALL USPLO (T,UMAT,201,NI,1,13HSYSTEM INPUTS,13,4HTIME,4
00160      1,5HINPUT,5,RANGE,10H1234567890,IOPT,IER)
00161      CALL UERTST (IER,6HUSPLO )
00162      135     IF (IPLOT.LE.0) GO TO 140
00163      C***** PLOT STATE VARIABLES OR OUTPUTS *****
00164      WRITE (5,33)
00165      READ (5,34) III
00166      CALL USPLO (T,XMAT,201,NI,1,15HTIME SIMULATION,15,4HTIME,4
00167      1,8HRESPONSE,8,RANGE,10H1234567890,IOPT,IER)
00168      CALL UERTST (IER,6HUSPLO )

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00169          GO TO 170
00170      140  DO 160 J=1,N
00171          DO 150 I=1,NP
00172          VEC(I)=XMAT(I,J)
00173      150  CONTINUE
00174          WRITE (5,33)
00175          READ (5,34) III
00176          CALL USPLD (T,VEC,201,NP,1,1,15HTIME SIMULATION,15,4HTIME,4
00177      1,8HRESPONSE,8,RANGE,1HX,IOPT,IER)
00178          CALL UERTST (IER,6HUSPLO )
00179      160  CONTINUE
00180      170  WRITE (5,27)
00181      27  FORMAT (1X,28HWISH TO REPEAT THE PLOTTING?)
00182          READ (5,*) K1
00183          IF (K1.GT.0) GO TO 115
00184          WRITE (5,28)
00185      28  FORMAT (1X,28HWISH TO EXIT FROM THIS MODE?)
00186          READ (5,*) K2
00187          IF (K2.LE.0) GO TO 180
00188          WRITE (5,29)
00189      29  FORMAT (1X,27(1H*),18H EXITING MODE 4 ,25(1H*))
00190          GO TO 200
00191      190  WRITE (5,31) IND,IER,K
00192      31  FORMAT (1X,4HIND=,I2,4HIER=,I3,51HCHECK INSTRUCTIONS FOR DIAGNOSTI
00193      1C MESSAGES ON DVERK,/,1X,28HPROBLEM ON ITERATION NUMBER ,I3)
00194          GO TO 53
00195      200  RETURN
00196          END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003     SUBROUTINE UEVAL(INOPT,AMP,SLOPE,U,NI,TEND)
00004 C-Function: Evaluates the input forcing functions.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: -   Output Unit: -
00008 C           Storage Unit(s): -
00009 C-Random Access Files: -
00010     INTEGER INOPT(NI)
00011     REAL AMP(NI),SLOPE(NI),U(NI),TEND
00012     DO 10 I=1,NI
00013     GO TO (1,2,3,4),INOPT(I)
00014     1   U(I)=0.000000
00015     GO TO 10
00016     2   U(I)=AMP(I)
00017     GO TO 10
00018     3   U(I)=SLOPE(I)*TEND
00019     GO TO 10
00020     4   IF (TEND.LE.(AMP(I)/SLOPE(I))) GO TO 3
00021     GO TO 2
00022 10    CONTINUE
00023     RETURN
00024     END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003     SUBROUTINE FCN(NS,T,X,XPRIME)
00004 C-Function: Evaluates x' fuctions for use by IMSL routine DVERK.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: -   Output Unit: -
00008 C           Storage Unit(s): -
00009 C-Random Access Files: -
00010     REAL X(NS),XPRIME(NS),ATIL(10,10),CONST(10)
00011     COMMON/DIF/ATIL,CONST
00012     DO 10 I=1,NS
00013     XPRIME(I)=ATIL(I,1)*X(1)+ATIL(I,2)*X(2)+ATIL(I,3)*X(3)+ATIL(I,4)*X
00014     1(4)+ATIL(I,5)*X(5)+ATIL(I,6)*X(6)+ATIL(I,7)*X(7)+ATIL(I,8)*X(8)+AT
00015     2IL(I,9)*X(9)+ATIL(I,10)*X(10)+CONST(I)
00016     10  CONTINUE
00017     RETURN
00018     END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003     SUBROUTINE MODE5
00004 C-Function: Main routine for Component Modification
00005 C-IMSL routines called: UERSET,USWFM.
00006 C-Spectral Assignment routines: CGRAD,CCOST,SEARCH,DSPLAY.
00007 C-Logical devices; Input Unit: 5 Output Unit: 5
00008 C- Storage Unit(s): IU=20,IU=20+J for J=1,NS,IUT=20+NS+1
00009 C-Random Access Files: SYSTEM.DAT,FORxx.DAT where xx=20+J ,CURRNT.DAT
00010     REAL AL(10,10),G(10,10)
00011     REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00012     REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00013     REAL A(10,10),B(10,10),C(10,10)
00014     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00015     COMMON/AUG/F,AHAT/EIG/LRE,LIM/PAR/AL/GR/G
00016     COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00017     COMMON/COMP/IROW,ICOL,F1,F2
00018     EXTERNAL CCOST,CGRAD
00019     CALL UERSET(3,LEVOLD)
00020     IU=20
00021     READ (IU'1) NS,NI,NO,IDGT,ZERO
00022     READ (IU'2) ((A(II,IJ),IJ=1,NS),II=1,NS)
00023     READ (IU'3) ((B(II,IJ),IJ=1,NI),II=1,NS)
00024     DO 10 J=1,NS
00025     IU=20+J
00026     OPEN (ACCESS='RANDOM',RECORD SIZE=202
00027     1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00028     READ (IU'1) LRE(J),LIM(J)
00029     10 CONTINUE
00030     IUT=20+NS+1
00031     OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=102
00032     1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00033     READ (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00034     READ (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00035     READ (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00036     READ (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00037     CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)      !***
00038     C CALL USWFM (10HMATRIX XX :,10,XX,10,NI,NS,4)      !***
00039     C CALL USWFM (14HGAIN MATRIX F :,14,F,10,NI,NS,4)      !***
00040     C CALL USWFM (12HMATRIX AHAT :,12,AHAT,10,NS,NS,4)      !***
00041     DO 20 II=1,NS
00042     DO 20 IJ=1,NS
00043     AL(II,IJ)=V(II,IJ)
00044     20 CONTINUE
00045     WRITE (5,1)
00046     1 FORMAT (1X,22(1H*),28H MODE 5:COMPONENT REDUCTION ,20(1H*),//,
00047     11X,52HENTER THE COORDINATES OF THE COMPONENT TO BE REDUCED,/,
00048     21X,32HROW=--,COLUMN=--(BOTH INTEGERS):)
00049     READ (5,*) IROW,ICOL
00050     WRITE (5,2)
00051     2 FORMAT (1X,39HSET DESIRED WEIGHTS,DEFAULT VALUES ARE:./,
00052     11X,11HF1=F2=1.000,/,1X,15HWISH TO CHANGE?)
00053     READ (5,*) KK
00054     F1=FLOAT(1)
00055     F2=F1
00056     IF (KK.LE.0) GO TO 30

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00057      WRITE (5,3)
00058      3  FORMAT (1X,17HENTER NEW VALUES:)
00059      READ (5,*) F1,F2
00060      30  CALL CCOST(CJ)
00061      WRITE (5,4) CJ
00062      4  FORMAT (1X,5HCOST=,E15.6)
00063      CALL CGRAD
00064      CALL SEARCH(CJ,CCOST,CGRAD,5)
00065      WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00066      WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00067      WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00068      WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00069      CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)
00070      WRITE (5,902)
00071      902 FORMAT (1X,44HWISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
00072      READ (5,*) KS
00073      IF (KS.LE.0) GO TO 903
00074      CALL DSPLAY (NS,ZERO)
00075      903 CONTINUE
00076      C  CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)
00077      CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)
00078      C  CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)
00079      RETURN
00080      END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE CCOST(CJ)
00004 C-Function: Calculates the COST function for component modification.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: 5
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00011 REAL W(10,10),V(10,10),VINV(10,10),AL(10,10)
00012 REAL A(10,10),B(10,10),C(10,10)
00013 COMMON/SYS/A,B,C,ZERO, IDGT,NS,NI,NO
00014 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00015 COMMON/COMP/IROW,ICOL,F1,F2/EIG/LRE,LIM/PAR/AL
00016 ICOL1=ICOL+1
00017 CJ1=F1*V(IROW,ICOL)**2
00018 IF (ABS(LIM(ICOL)).GT.ABS(ZERO)) CJ1=CJ1+F1*V(IROW,ICOL1)**2
00019 CJ2=FLOAT(0)
00020 N=1
00021 10 N1=N+1
00022 DO 100 M=1,NS
00023 IF (N.EQ.ICOL.AND.M.EQ.IROW) GO TO 100
00024 CJ2=CJ2+((V(M,N)-AL(M,N))**2)*F2
00025 IF(ABS(LIM(N)).GT.ABS(ZERO)) CJ2=CJ2+(V(M,N1)-AL(M,N1))**2*F2
00026 100 CONTINUE
00027 N=N+1
00028 IF (ABS(LIM(N)).GT.ABS(ZERO)) N=N+1
00029 IF (N.LE.NS) GO TO 10
00030 CJ=CJ1+CJ2
00031 WRITE (5,1) CJ1,CJ2
00032 1 FORMAT (20X,4H J1=,E15.6,5X,4H J2=,E15.6)
00033 RETURN
00034 END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003 SUBROUTINE CGRAD
00004 C-Function: Calculates the GRADIENT for component modification.
00005 C-IMSL routines called: USWFM
00006 C-Spectral Assignment routines: PVP,DBNORM.
00007 C-Logical devices; Input Unit: - Output Unit: 5
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL G(10,10),PJ1(10,10),PJ2(10,10)
00011 REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00012 REAL W(10,10),V(10,10),VINV(10,10),AL(10,10)
00013 REAL A(10,10),B(10,10),C(10,10)
00014 REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
00015 COMMON/SYS/A,B,C,ZERO,IGDT,NS,NI,NO
00016 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/AUX/AUX1,AUX2,AUX3
00017 COMMON/COMP/IROW,ICOL,F1,F2/EIG/LRE,LIM/PAR/AL/GR/G/PJ/PJ1,PJ2
00018 ICOL1=ICOL+1
00019 J=1
00020 10 J1=J+1
00021 DO 105 I=1,NI
00022 KI=I
00023 KJ=J
00024 CALL PVP(KI,KJ)
00025 IF (ICOL.NE.J) GO TO 14
00026 PJ1(I,J)=2*F1*V(IROW,ICOL)*AUX1(IROW,ICOL)
00027 IF (ABS(LIM(ICOL)).LE.ABS(ZERO)) GO TO 15
00028 C CALL USWFM(8HPV/PXIJ:,8,AUX1,10,NS,NS,4) !**
00029 C CALL USWFM(10HPV/PXIJ+1:,10,AUX2,10,NS,NS,4) !**
00030 PJ1(I,J)=PJ1(I,J)+2*F1*V(IROW,ICOL)*AUX1(IROW,ICOL)
00031 PJ1(I,J1)=2*F1*(V(IROW,ICOL)*AUX2(IROW,ICOL)+V(IROW,ICOL)*
00032 1AUX2(IROW,ICOL))
00033 GO TO 15
00034 14 PJ1(I,J)=FLOAT(0)
00035 IF (ABS(LIM(ICOL)).GT.ABS(ZERO)) PJ1(I,J1)=FLOAT(0)
00036 15 PJ2(I,J)=FLOAT(0)
00037 IF (ABS(LIM(J)).GT.ABS(ZERO)) PJ2(I,J1)=FLOAT(0)
00038 DO 100 M=1,NS
00039 IF (J.EQ.ICOL.AND.M.EQ.IROW) GO TO 100
00040 PJ2(I,J)=PJ2(I,J)+(AL(M,J)-V(M,J))*AUX1(M,J)
00041 IF (ABS(LIM(J)).LE.ABS(ZERO)) GO TO 100
00042 PJ2(I,J)=PJ2(I,J)+(AL(M,J1)-V(M,J1))*AUX1(M,J1)
00043 PJ2(I,J1)=PJ2(I,J1)+(AL(M,J)-V(M,J))*AUX2(M,J)+
00044 1(AL(M,J1)-V(M,J1))*AUX2(M,J1)
00045 100 CONTINUE
00046 PJ2(I,J)=2*F2*PJ2(I,J)
00047 IF (ABS(LIM(J)).GT.ABS(ZERO)) PJ2(I,J1)=2*F2*PJ2(I,J1)
00048 105 CONTINUE
00049 IF (ABS(LIM(J)).GT.ABS(ZERO)) J=J+1
00050 J=J+1
00051 IF (J.LE.NS) GO TO 10
00052 DO 110 II=1,NI
00053 DO 110 IJ=1,NS
00054 G(II,IJ)=PJ1(II,IJ)+PJ2(II,IJ)
00055 110 CONTINUE
00056 C CALL USWFM(11HMATRIX [G]:,11,G,10,NI,NS,4) !**
00057 CALL DBNORM(NI,NS)
00058 CALL USWFM(16HGRADIENT MATRIX:,16,G,10,NI,NS,4) !**
00059 RETURN
00060 END

```



```

00001 C*****
00002 C*****
00003     SUBROUTINE PVP(KI,KJ)
00004 C-Function: Returns p[V]/p[X]ij .
00005 C-IMSL routines called: (USWFM).
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit:   -   Output Unit:   (5)
00008 C-   Storage Unit(s): IU=20+KJ ,KJ specified by CALL statement.
00009 C-Random Access Files: FORxx.DAT where xx=20+KJ.
00010     REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
00011 C NULL SPACE ARRAYS
00012     REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
00013     REAL NLC(10,20),PLC(10,20),MLC(10,20)
00014     REAL STAR(20,20),QL(10,20),RL(10,20)
00015 C MODE 3 ARRAYS
00016     REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00017     REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00018     REAL A(10,10),B(10,10),C(10,10)
00019     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00020     COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/EIG/LRE,LIM
00021     COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00022     COMMON/AUX/AUX1,AUX2,AUX3
00023     IU=20+KJ
00024     IF (ABS(LIM(KJ)).GT.ABS(ZERO)) GO TO20
00025     READ (IU'3) ((NL(IJ,I),I=1,NI),I=1,NS)
00026     DO 10 I=1,NS
00027     DO 10 J=1,NI
00028     AUX1(I,J)=FLOAT(0)
00029     IF (J.EQ.KJ) AUX1(I,J)=NL(I,KI)
00030     AUX2(I,J)=FLOAT(0)
00031     10 CONTINUE
00032     GO TO 30
00033     20 NI2=NI*2
00034     READ (IU'6) ((QL(IJ,I),I=1,NI2),I=1,NS)
00035     READ (IU'7) ((RL(IJ,I),I=1,NI2),I=1,NS)
00036 C     CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4)     !**
00037 C     CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4)     !**
00038     KIN=KI+NI
00039     KJ1=KJ+1
00040     DO 30 I=1,NS
00041     DO 30 J=1,NI
00042     AUX1(I,J)=FLOAT(0)
00043     AUX2(I,J)=FLOAT(0)
00044     IF (J.EQ.KJ) AUX1(I,J)=QL(I,KI)
00045     IF (J.EQ.KJ1) AUX1(I,J)=RL(I,KI)
00046     IF (J.EQ.KJ) AUX2(I,J)=QL(I,KIN)
00047     IF (J.EQ.KJ1) AUX2(I,J)=RL(I,KIN)
00048     30 CONTINUE
00049 C     WRITE (5,1) KI,KJ     !**
00050 C 1     FORMAT (1X,'I=',I2,'J=',I2)     !**
00051 C     CALL USWFM(BHPV/PXIJ:,8,AUX1,10,NS,NS,4)     !**
00052 C     CALL USWFM(10HPV/PXIJ+1:,10,AUX2,10,NS,NS,4)     !**
00053     RETURN
00054     END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE MODE6
00004 C-Function: Main routine for Gain Modification.
00005 C-IMSL routines called: UERSET,USWFM.
00006 C-Spectral Assignment routines: GCDST,GGRAD,SEARCH,DSPLAY.
00007 C-Logical devices; Input Unit: 5 Output Unit: 5
00008 C- Storage Unit(s): IU=20,IUT=20+NS+1
00009 C-Random Access Files: SYSTEM.DAT,CURRNT.DAT .
00010 C GRADIENT ARRAYS
00011 REAL AL(10,10),G(10,10),AUX1(10,10),AUX2(10,10),AUX3(10,10)
00012 C NULL SPACE ARRAYS
00013 REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
00014 REAL NLC(10,20),PLC(10,20),MLC(10,20)
00015 REAL STAR(20,20),QL(10,20),RL(10,20)
00016 C MODE 3 ARRAYS
00017 REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00018 REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00019 REAL A(10,10),B(10,10),C(10,10)
00020 COMMON/SYS/A,B,C,ZERO, IDGT,NS,NI,NO
00021 COMMON/AUG/F,AHAT/EIG/LRE,LIM/PAR/AL/GR/G
00022 COMMON/VEC/VA,F,X,WJ,W,XX,V,VINV
00023 COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00024 COMMON/AUX/AUX1,AUX2,AUX3
00025 EXTERNAL GCOST,GGRAD
00026 CALL UERSET(3,LEVOLD)
00027 IU=20
00028 READ (IU*1) NS,NI,NO, IDGT,ZERO
00029 READ (IU*2) ((A(II,IJ),IJ=1,NS),II=1,NS)
00030 READ (IU*3) ((B(II,IJ),IJ=1,NI),II=1,NS)
00031 IUT=20+NS+1
00032 OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=102
00033 1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00034 READ (IUT*1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00035 READ (IUT*2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00036 READ (IUT*4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00037 READ (IUT*5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00038 CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4) !!!
00039 C CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4) !!!
00040 CALL USWFM (14HGAIN MATRIX F :,14,F,10,NI,NS,4) !!!
00041 C CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4) !!!
00042 DO 10 II=1,NI
00043 DO 10 IJ=1,NS
00044 AL(II,IJ)=FLOAT(1)
00045 10 CONTINUE
00046 WRITE (5,1)
00047 1 FORMAT (1X,22(1H*),23H MODE 6:GAIN REDUCTION ,25(1H*),//,
00048 11X,22HSET ALPHA PARAMETERS :/,1X,20HDEFAULT VALUES ARE :)
00049 CALL USWFM (17HGAIN PARAMETERS :,17,AL,10,NI,NS,4) !!!
00050 WRITE (5,2)
00051 2 FORMAT (1X,15HWISH TO CHANGE:)
00052 READ (5,*) KK
00053 IF (KK.LE.0) GO TO 20
00054 WRITE (5,4)
00055 4 FORMAT (1X,17HENTER NEW VALUES:)
00056 READ (5,*) ((AL(II,IJ),IJ=1,NS),II=1,NI)
```

```

00057      20      CALL GCOST(CJ)
00058      WRITE (5,3) CJ
00059      3      FORMAT (1X,5HCOST=,E15.6)
00060      CALL GGRAD
00061      CALL SEARCH(CJ,GCOST,GGRAD,6)
00062      WRITE (IUT*1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00063      WRITE (IUT*2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00064      WRITE (IUT*4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00065      WRITE (IUT*5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00066      CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)
00067      WRITE (5,902)
00068      902      FORMAT (1X,44HNISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
00069      READ (5,*) KS
00070      IF (KS.LE.0) GO TO 903
00071      CALL DSPLAY (NS,ZERO)
00072      903      CONTINUE
00073      C      CALL USWFM (10HMATRIX XX :,10,XX,10,NI,NS,4)
00074      CALL USWFM (14HGAIN MATRIX F :,14,F,10,NI,NS,4)
00075      C      CALL USWFM (12HMATRIX AHAT :,12,AHAT,10,NS,NS,4)
00076      RETURN
00077      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003 SUBROUTINE SEARCH(CJ,COST,GRAD,MODE)
00004 C-Function: Inter-active Gradient Search Routine.
00005 C-IMSL routines called: UERSET,UERTST,LINV2F,(USWFM),
00006 C-Spectral Assignment routines: GAIN,COST,DESIGN,SENS,GRAD,TRAN.
00007 C-Logical devices: Input Unit: 5 Output Unit: 5
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 C GRADIENT ARRAYAS
00011 REAL AL(10,10),G(10,10),U(10,10),WKAREA(100)
00012 C NULL SPACE ARRAYS
00013 REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
00014 REAL NLC(10,20),PLC(10,20),MLC(10,20)
00015 REAL STAR(20,20),QL(10,20),RL(10,20)
00016 C MODE 3 ARRAYS
00017 REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00018 REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00019 REAL A(10,10),B(10,10),C(10,10)
00020 COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00021 COMMON/AUG/F,AHAT/EIG/LRE,LIM/PAR/AL/GR/G/LEG/U
00022 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00023 COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00024 CALL UERSET(3,LEVOLD)
00025 IFL=0
00026 KN=1
00027 N=1
00028 D=0.01
00029 DMIN=ZERO
00030 10 WRITE (5,1) N,D,DMIN
00031 1 FORMAT (1X,46HGRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS://
00032 1,1X,19HDefault values are://,1X,13H# of steps,N=,I3,3X,12Hstep siz
00033 2e,d=,E15.6,3X,5Hdmin=,E15.6,/,1X,15HWish to change?)
00034 READ (5,*) IUP
00035 IF (IUP.LE.0) GO TO 20
00036 WRITE (5,2)
00037 2 FORMAT (1X,17Henter new values:)
00038 READ (5,*) N,D,DMIN
00039 20 IN=1
00040 30 DO 40 II=1,NI
00041 DO 40 IJ=1,NS
00042 XX(II,IJ)=XX(II,IJ)-D*G(II,IJ)
00043 40 CONTINUE
00044 CALL DESIGN
00045 CALL GAIN
00046 IF (MODE.NE.7) GO TO 49
00047 IDG=IDGT
00048 CALL LINV2F (V,NS,10,U,IDG,WKAREA,IER)
00049 CALL UERTST(IER,6HLINV2F)
00050 C CALL USWFM (10HMATRIX UT:,10,U,10,NS,NS,4) ***
00051 CALL tran(U,NS,NS)
00052 C CALL USWFM (10HMATRIX U :,10,U,10,NS,NS,4) ***
00053 CALL SENS
00054 49 CALL COST(CJNEW)
00055 WRITE (5,7) CJNEW ***
00056 7 FORMAT (1X,'NEW COST=',E15.6) ***

```

ORIGINAL PAGE 13  
OF POOR QUALITY

```

00057      IF (CJNEW.GE.CJ) GO TO 50
00058      IF (IN.GE.N) GO TO 100
00059      CJ=CJNEW
00060      IN=IN+1
00061      KN=KN+1
00062      GO TO 30
00063  50     DO 60 II=1,NI
00064      DO 60 IJ=1,NS
00065      XX(II,IJ)=XX(II,IJ)+D*G(II,IJ)
00066  60     CONTINUE
00067      IF (KN.EQ.1) GO TO 70
00068      WRITE (5,3) KN,D
00069  3     FORMAT (1X,I3,38H Steps with present gradient and dmin=,E15.6
00070      1,10Hwere taken,/,1X,23HLAST STEP NOT ACCEPTED!)
00071      KN=1
00072      CALL GRAD
00073      GO TO 30
00074  70     DH=D/2
00075      WRITE (5,8) DH
00076  8     FORMAT (1X,23HLAST STEP NOT ACCEPTED!/,
00077      1,1X,21HSTEP SIZE REDUCED TO:,E15.6)
00078      IF (DH.LT.DMIN) GO TO 80
00079      D=DH
00080      GO TO 30
00081  80     WRITE (5,4)
00082  4     FORMAT (1X,36HYou are in d/2 neighborhood of Jmin!)
00083      IFL=1
00084  100    CALL DESIGN
00085      CALL GAIN
00086      WRITE (5,5) CJNEW
00087  5     FORMAT (1X,14HCost Function=,E15.6)
00088  C     CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)      ***
00089  C     CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)    ***
00090  C     CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4) ***
00091  C     CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4) ***
00092      IF (IFL.EQ.1) GO TO 90
00093      WRITE (5,6)
00094  6     FORMAT (1X,28HWish to continue the search?)
00095      READ (5,*) KK
00096      IF (KK.LE.0) GO TO 90
00097      CJ=CJNEW
00098      GO TO 10
00099  90     RETURN
00100      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE GCOST(CJ)
00004 C-Function: Calculates the COST function for Gain Reduction.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: (5)
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL A(10,10),B(10,10),C(10,10)
00011 REAL AL(10,10),F(10,10),AHAT(10,10)
00012 COMMON/AUG/F,AHAT/PAR/AL
00013 COMMON/SYS/A,B,C,ZERO,IOGT,NS,NI,NO
00014 CJ=FLOAT(0)
00015 DO 10 I=1,NI
00016 DO 10 J=1,NS
00017 CJ=CJ+AL(I,J)*(F(I,J)**2)
00018 10 CONTINUE
00019 C WRITE (5,1) CJ ***
00020 C 1 FORMAT (1X,5HCOST=,E15.6) ***
00021 RETURN
00022 END
```

```

00001 C*****
00002 C*****
00003     SURROUTINE GGRAD
00004 C-Function: Calculates the Gradient for Gain Reduction.
00005 C-IMSL routines called: UERTST,USWFM,LINV2F.
00006 C-Spectral Assignment routines: DBNORM,PFX.
00007 C-Logical devices; Input Unit: - Output Unit: 5
00008 C- Storage Unit(s): IU=20+J for J=1,NS.
00009 C-Random Access Files: FQRxx.DAT where xx=20+J .
00010 C GRADIENT ARRAYS
00011     REAL AL(10,10),G(10,10),AUX1(10,10),AUX2(10,10),AUX3(10,10)
00012 C NULL SPACE ARRAYS
00013     REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
00014     REAL NLC(10,20),PLC(10,20),MLC(10,20)
00015     REAL STAR(20,20),QL(10,20),RL(10,20)
00016 C AUX. ARRAYS
00017     REAL WKAREA(130)
00018 C MODE 3 ARRAYS
00019     REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00020     REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00021     REAL A(10,10),B(10,10),C(10,10)
00022     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00023     COMMON/AUG/F,AHAT/EIG/LRE,LIM/PAR/AL/GR/G
00024     COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00025     COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00026     COMMON/AUX/AUX1,AUX2,AUX3
00027 C     WRITE (5,1)                                !***
00028 C 1     FORMAT (1X,'SUBROUTINE GGRAD+++++') !***
00029     CALL LINV2F (V,NS,10,VINV,IDGT,WKAREA,IER)
00030     CALL UERTST (IER,6HLINV2F)
00031 C     CALL USWFM (12HMATRIX VINV:,12,VINV,10,NS,NS,4)  !***
00032     J=1
00033 10    CONTINUE
00034     IRS=202
00035     IU=J+20
00036     OPEN (ACCESS='RANDOM',RECORD SIZE=IRS,UNIT=IU
00037           1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00038 C***** Is Lambda-J real? *****
00039     READ (IU'1) LRE(J),LIM(J)
00040     IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 30
00041 C***** Find partials of J wrt elements of [XX], real case**
00042     READ (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
00043     READ (IU'4) ((ML(II,IJ),IJ=1,NI),II=1,NI)
00044 C     CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4)  !***
00045 C     CALL USWFM (10HMATRIX ML:,10,ML,10,NI,NI,4)  !***
00046     GO TO 15
00047 C***** Find complex partials *****
00048 30    IS=NS+NI
00049     NI2=2*NI
00050     NS2=2*NS
00051     INS=NS+1
00052     READ (IU'3) ((NLC(II,IJ),IJ=1,IS),II=1,NS)
00053     READ (IU'4) ((PLC(II,IJ),IJ=1,IS),II=1,NS)
00054     READ (IU'5) ((MLC(II,IJ),IJ=1,IS),II=1,NI)
00055     READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
00056     READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00057 C      CALL USWFM (11HMATRIX NLC:,11,NLC,10,NS,IS,4)      !!!
00058 C      CALL USWFM (11HMATRIX PLC:,11,PLC,10,NS,IS,4)      !!!
00059 C      CALL USWFM (11HMATRIX MLC:,11,MLC,10,NI,IS,4)      !!!
00060 C      CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4)         !!!
00061 C      CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4)         !!!
00062   15    DO 100 I=1,NI
00063          G(I,J)=FLOAT(0)
00064          KJ=J
00065          KI=I
00066          CALL PFX(KI,KJ,IFLAG)
00067          IF (IFLAG.NE.0) GO TO 150
00068          DO 100 IP=1,NI
00069          DO 100 IQ=1,NS
00070          G(I,J)=G(I,J)+2*AL(IP,IQ)*F(IP,IQ)*AUX3(IP,IQ)
00071 C      WRITE (5,2) IP,IQ,I,J,G(I,J)                          !!!
00072 C  2      FORMAT (20X,'PF',I2,I2,'/X',I2,I2,' =',E15.6,'PARTIAL SUMS')!!!
00073   100    CONTINUE
00074          IF (IFLAG.EQ.1) J=J+1
00075          IF (J.GE.NS) GO TO 200
00076          J=J+1
00077          GO TO 10
00078   150    JD=J+1
00079          DO 70 IP=1,NI
00080          DO 70 IQ=1,NS
00081          G(I,J)=G(I,J)+2*AL(IP,IQ)*F(IP,IQ)*W(IP,IQ)
00082   70    CONTINUE
00083          G(I,JD)=FLOAT(0)
00084          DO 75 IP=1,NI
00085          DO 75 IQ=1,NS
00086          G(I,JD)=G(I,JD)+2*AL(IP,IQ)*F(IP,IQ)*AUX3(IP,IQ)
00087   75    CONTINUE
00088          GO TO 100
00089 C***** Print [G], then find [G]/[G]! *****
00090   200    CONTINUE
00091 C      CALL USWFM (11HMATRIX [G]:,11,G,10,NI,NS,4)          !!!
00092          CALL DBNORM(NI,NS)
00093          CALL USWFM (16HGradient matrix:,16,G,10,NI,NS,4)   !!!
00094          RETURN
00095          END

```



```

00001 C*****
00002 C*****
00003 SUBROUTINE INSTEP
00004 C-Function: Called by PFX calculates [AUX3]=([AUX1]-[AUX2])*[VINV]
00005 C-IMSL routines called: UERTST,VMULFF,(USWFM).
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: (5)
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
00011 REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
00012 REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00013 REAL A(10,10),B(10,10),C(10,10)
00014 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/AUG/F,AHAT
00015 COMMON/AUX/AUX1,AUX2,AUX3
00016 COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00017 C CALL USWFM (7H[AUX1]:,7,AUX1,10,NI,NS,4) !**
00018 C CALL USWFM (7H[AUX2]:,7,AUX2,10,NS,NS,4) !**
00019 CALL VMULFF(F,AUX2,NI,NS,NS,10,10,AUX3,10,IER)
00020 CALL UERTST(IER,6HVMULFF)
00021 C CALL USWFM (7H[AUX3]:,7,AUX3,10,NI,NS,4) !**
00022 DO 10 II=1,NI
00023 DO 10 IJ=1,NS
00024 AUX2(II,IJ)=AUX1(II,IJ)-AUX3(II,IJ)
00025 10 CONTINUE
00026 CALL VMULFF(AUX2,VINV,NI,NS,NS,10,10,AUX3,10,IER)
00027 CALL UERTST(IER,6HVMULFF)
00028 C CALL USWFM (7H[AUX3]:,7,AUX3,10,NI,NS,4) !**
00029 RETURN
00030 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE DBNORM(NI,NS)
00004 C-Function: Returns a normalized NixNS matrix in itself.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: -
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL G(10,10),NORM
00011 COMMON/GR/G
00012 NORM=FLOAT(0)
00013 DO 10 I=1,NI
00014 DO 10 J=1,NS
00015 NORM=NORM+G(I,J)**2
00016 10 CONTINUE
00017 NORM=SQRT(NORM)
00018 DO 20 I=1,NI
00019 DO 20 J=1,NS
00020 G(I,J)=G(I,J)/NORM
00021 20 CONTINUE
00022 RETURN
00023 END
```

```

00001 C*****
00002 C*****
00003 SUBROUTINE DESIGN
00004 C-Function: Given a Designator matrix [X], calculates [V].
00005 C-IMSL routines called: UERTST,VMULFF,(USWFM).
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: (5)
00008 C- Storage Unit(s): IU=20+J
00009 C-Random Access Files: FORxx.DAT where xx=20+J for J=1,NS.
00010 C NULL SPACE ARRAYS
00011 REAL ML(10,10),NL(10,10)
00012 REAL NLC(10,20),PLC(10,20),MLC(10,20)
00013 REAL STAR(20,20),QL(10,20),RL(10,20)
00014 C MODE 3 ARRAYS
00015 REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00016 REAL W(10,10),V(10,10),VINV(10,10)
00017 REAL A(10,10),B(10,10),C(10,10)
00018 COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00019 COMMON/EIG/LRE,LIM
00020 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00021 COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00022 C WRITE (5,1) ***
00023 C 1 FORMAT (1X,'SUBROUTINE DESIGN ++++++') !**
00024 IRS=202
00025 J=1
00026 10 IU=J+20
00027 OPEN (ACCESS='RANDOM',RECORD SIZE=IRS,UNIT=IU
00028 1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00029 C***** Is Lambda-J real? *****
00030 READ (IU'1) LRE(J),LIM(J)
00031 IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 30
00032 C***** Find real VA=J-th column of [V] *****
00033 READ (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
00034 C CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4) !**
00035 DO 20 IV=1,NI
00036 X(IV)=XX(IV,J)
00037 20 CONTINUE
00038 C***** Find VA=[NL]*X and put it in J-th column of [V] *****
00039 CALL VMULFF (NL,X,NS,NI,1,10,20,VA,20,IER)
00040 CALL UERTST (IER,6HVMULFF)
00041 C CALL USWFV (10HVECTOR VA:,10,VA,NS,1,4) !**
00042 DO 25 IV=1,NS
00043 V(IV,J)=VA(IV)
00044 25 CONTINUE
00045 29 IF (J.GE.NS) GO TO 100
00046 J=J+1
00047 GO TO 10
00048 C***** Find complex VA's *****
00049 30 INI=NI+1
00050 NI2=2*NI
00051 NS2=2*NS
00052 INS=NS+1
00053 READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
00054 READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
00055 C CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4) !**
00056 C CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4) !**

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00057      IC=J+1
00058 C***** Form [STAR] and double length X *****
00059      DO 135 II=1,NS
00060      DO 135 IJ=1,NI2
00061      STAR(II,IJ)=QL(II,IJ)
00062      135 CONTINUE
00063      DO 140 II=INS,NS2
00064      DO 140 IJ=1,NI2
00065      IDUM=II-NS
00066      STAR(II,IJ)=RL(IDUM,IJ)
00067      140 CONTINUE
00068 C      CALL USWFM (12HMATRIX STAR:,12,STAR,20,NS2,NI2,4) !**
00069      DO 40 IV=1,NI
00070      X(IV)=XX(IV,J)
00071      40 CONTINUE
00072      DO 50 IV=INI,NI2
00073      IVDUM=IV-NI
00074      X(IV)=XX(IVDUM,IC)
00075      50 CONTINUE
00076 C      CALL USWFV (10HVECTOR XT:,10,X,NI2,1,4) !**
00077 C***** Find VA=[*].X and partition it to [V]j, [V]j+1 *****
00078      CALL VMULFF (STAR,X,NS2,NI2,1,20,20,VA,20,IER)
00079      CALL UERTST (IER,6HVMULFF)
00080 C      CALL USWFV (10HVECTOR VA:,10,VA,NS2,1,4) !**
00081      DO 60 IV=1,NS
00082      V(IV,J)=VA(IV)
00083      IVD=IV+NS
00084      V(IV,IC)=VA(IVD)
00085      60 CONTINUE
00086      J=IC
00087      GO TO 29
00088 C***** Print [V] *****
00089      100 CONTINUE
00090 C      CALL USWFM (11HMATRIX [V]:,11,V,10,NS,NS,4) !**
00091 C      WRITE (5,2) !**
00092 C 2      FORMAT (1X,'EXITING DESIGN =====') !**
00093      RETURN
00094      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003       SUBROUTINE PFX(I,J,IFLAG)
00004 C-Function: Returns p[F]/p[x]ij .
00005 C-IMSL routines called: UERTST,LLSQF,VMULFF,(USWFM).
00006 C-Spectral Assignment routines: INSTEP.
00007 C-Logical devices; Input Unit: -   Output Unit:   (5)
00008 C-       Storage Unit(s): -
00009 C-Random Access Files: -
00010 C GRADIENT ARRAYAS
00011       REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
00012 C NULL SPACE APRAYS
00013       REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
00014       REAL NLC(10,20),PLC(10,20),MLC(10,20)
00015       REAL STAR(20,20),QL(10,20),RL(10,20)
00016 C AUX. ARRAYS
00017       REAL WKAREA(130),H(20)
00018 C MODE 3 ARRAYS
00019       REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00020       REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00021       INTEGER IPA(20)
00022       REAL A(10,10),B(10,10),C(10,10)
00023       COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00024       COMMON/EIG/LRE,LIM
00025       COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00026       COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00027       COMMON/AUX/AUX1,AUX2,AUX3
00028 C***** Is Lambda=J real? *****
00029       IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 30
00030 C***** Find partials of J wrt elements of [XX], real case**
00031       DO 15 II=1,NI
00032       DO 15 IJ=1,NS
00033         AUX1(IJ,II)=FLOAT(0)
00034         IF (IJ.EQ.J) AUX1(IJ,II)=-ML(II,I)
00035     15 CONTINUE
00036       DO 20 II=1,NS
00037       DO 20 IJ=1,NS
00038         AUX2(IJ,II)=FLOAT(0)
00039         IF (IJ.EQ.J) AUX2(IJ,II)=NL(II,I)
00040     20 CONTINUE
00041 C       CALL USWFM (7H[AUX1]:,7,AUX1,10,NI,NS,4)           !**
00042 C       CALL USWFM (7H[AUX2]:,7,AUX2,10,NS,NS,4)           !**
00043       CALL INSTEP
00044       IFLAG=0
00045       GO TO 999
00046 C***** Find complex partials *****
00047     30 IS=NS+NI
00048       NI2=2*NI
00049       NS2=2*NS
00050       INS=NS+1
00051       JD=J
00052       JC=J+1
00053       INOW=I
00054 C***** FORM [STAR],[ALPHA],[BETA] *****
00055       DO 110 II=1,NS
00056       DO 110 IJ=1,NI2
00057         STAR(II,IJ)=QL(II,IJ)
00058         IDUM=II+NS
00059         STAR(IDUM,IJ)=RL(II,IJ)
00060     110 CONTINUE

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00061 C CALL USWFM (12HMATRIX STAR:,12,STAR,20,NS2,NI2,4) !**
00062 DO 135 II=1,NS
00063 DO 135 IJ=1,IS
00064 ALPHA(II,IJ)=NLC(II,IJ)
00065 135 CONTINUE
00066 DO 140 II=INS,NS2
00067 DO 140 IJ=1,IS
00068 IDUM=II-NS
00069 ALPHA(II,IJ)=-PLC(IDUM,IJ)
00070 140 CONTINUE
00071 C CALL USWFM (14HMATRIX ALPHA :,14,ALPHA,20,NS2,IS,4) !**
00072 DO 185 II=1,NS
00073 DO 185 IJ=1,IS
00074 BETA(II,IJ)=PLC(II,IJ)
00075 185 CONTINUE
00076 DO 190 II=INS,NS2
00077 DO 190 IJ=1,IS
00078 IDUM=II-NS
00079 BETA(II,IJ)=NLC(IDUM,IJ)
00080 190 CONTINUE
00081 C CALL USWFM (13HMATRIX BETA :,13,BETA,20,NS2,IS,4) !**
00082 50 CONTINUE
00083 DO 55 II=1,NS2
00084 E(II)=STAR(II,INOW)
00085 VA(II)=STAR(II,INOW)
00086 55 CONTINUE
00087 C CALL USWFV (19HI-th column of [*]:,19,E,NS2,1,4) !**
00088 C CALL USWFV (19HI-th column of [*]:,19,VA,NS2,1,4) !**
00089 CALL LLSQF (ALPHA,20,NS2,IS,E,-1.0,IS,X,H,IPA,IER)
00090 CALL UERTST (IER,6HLLSQF )
00091 C CALL USWFV (15HVECTOR [TM1]-I:,15,X,IS,1,4) !**
00092 C*** Form E=[MLC]*[TM1]i and put E in J-th column of [AUX1] ****
00093 CALL VMULFF (MLC,X,NI,IS,1,10,20,E,20,IER)
00094 CALL UERTST (IER,6HVMULFF)
00095 C CALL USWFV (10H[AUX1]-J :,10,E,NI,1,4) !**
00096 DO 180 IV=1,IS
00097 X(IV)=FLOAT(0)
00098 180 CONTINUE
00099 CALL LLSQF (BETA,20,NS2,IS,VA,-1.0,IS,X,H,IPA,IER)
00100 CALL UERTST (IER,6HLLSQF )
00101 C CALL USWFV (15HVECTOR [TM2]-I:,15,X,IS,1,4) !**
00102 C** Form VA=[MLC]*[TM1]i and put VA in J+1th column of [AUX1] ***
00103 CALL VMULFF (MLC,X,NI,IS,1,10,20,VA,20,IER)
00104 CALL UERTST (IER,6HVMULFF)
00105 C CALL USWFV (10H[AUX1]J+1:,10,VA,NI,1,4) !**
00106 DO 60 II=1,NI
00107 DO 60 IJ=1,NS
00108 AUX1(II,IJ)=FLOAT(0)
00109 IF (IJ.EQ.JD) AUX1(II,IJ)=E(II)
00110 IF (IJ.EQ.JC) AUX1(II,IJ)=VA(II)
00111 60 CONTINUE
00112 C CALL USWFM (8H[AUX1] :,8,AUX1,10,NI,NS,4) !**
00113 DO 70 II=1,NS
00114 DO 70 IJ=1,NS
00115 AUX2(II,IJ)=FLOAT(0)
00116 IF (IJ.EQ.JD) AUX2(II,IJ)=QL(II,INOW)
00117 IF (IJ.EQ.JC) AUX2(II,IJ)=RL(II,INOW)
00118 70 CONTINUE
00119 C CALL USWFM (8H[AUX2] :,8,AUX2,10,NS,NS,4) !**
00120 CALL INSTEP
00121 IF (INOW.NE.I) GO TO 999

```

```
00122      IFLAG=1
00123      DO 80 II=1,NI
00124      DO 80 IJ=1,NS
00125      W(II,IJ)=AUX3(II,IJ)
00126      80      CONTINUE
00127      INOW=I+NI
00128      GO TO 50
00129      999     RETURN
00130      END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002
00003 C*****
00004 SUBROUTINE MODE7
00005 C-Function: Main routine for Sensitivity Reduction.
00006 C-IMSL routines called: UERSET,UERTST,LINV2F,USWFM.
00007 C-Spectral Assignment routines: SEARCH,TRAN,SGRAD,SCOST,SENS.
00008 C-Logical devices; Input Unit: 5 Output Unit: 5
00009 C- Storage Unit(s): IU=20,IUT=20+NS+1,IU=20+J for J=1,NS.
00010 C-Random Access Files: SYSTEM.DAT,CURRNT.DAT,FORxx.DAT where xx=20+J.
00011 REAL WKAREA(130),U(10,10)
00012 C GRADIENT ARRAYAS
00013 REAL L(10),P(10),DAD(10,10),DBD(10,10),DAHD(10,10)
00014 C NULL SPACE ARRAYS
00015 REAL ML(10,10),NL(10,10)
00016 REAL NLC(10,20),PLC(10,20),MLC(10,20)
00017 REAL STAR(20,20),QL(10,20),RL(10,20)
00018 C MODE 3 ARRAYS
00019 REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00020 REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
00021 REAL A(10,10),B(10,10),C(10,10)
00022 COMMON/SYS/A,B,C,ZERO, IDGT,NS,NI,NO
00023 COMMON/AUG/F,AHAT/EIG/LRE,LIM/WET/L,P/GR/G/SEN/DAD,DBD,DAHD/LEG/U
00024 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00025 COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00026 EXTERNAL SCOST,SGRAD
00027 CALL UERSET(3,LEVOLD)
00028 IU=20
00029 READ (IU'1) NS,NI,NO, IDGT,ZERO
00030 READ (IU'2) ((A(II,IJ),IJ=1,NS),II=1,NS)
00031 READ (IU'3) ((B(II,IJ),IJ=1,NI),II=1,NS)
00032 IUT=20+NS+1
00033 OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=102
00034 1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00035 READ (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00036 READ (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00037 READ (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00038 READ (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00039 CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4) ***
00040 CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4) ***
00041 CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4) ***
00042 CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4) ***
00043 DO 30 J=1,NS
00044 IU=J+20
00045 IRS=202
00046 OPEN (ACCESS='RANDOM',RECORD SIZE=IRS,UNIT=IU
00047 1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
00048 READ (IU'1) LRE(J),LIM(J)
00049 30 CONTINUE
00050 WRITE (5,1)
00051 1 FORMAT (1X,20(1H*),30H MODE 7:SENSITIVITY REDUCTION ,20(1H*),//
00052 1,1X,23H Set weighting factors:,/
00053 2,1X,34H Eigenvalue weighting factors are:)
00054 DO 10 IV=1,NS
00055 L(IV)=FLOAT(1)
00056 WRITE (5,3) IV,L(IV)
00057 3 FORMAT (1X,2HL(,I2,2H)=,F15.6)
00058 10 CONTINUE

```



ORIGINAL PAGE IS  
OF POOR QUALITY.

```

00059      WRITE (5,5)
00060      5  FORMAT (1X,15HWish to change?)
00061      READ (5,*) KL
00062      IF (KL.LE.0) GO TO 11
00063      READ (5,*) (L(IV),IV=1,NS)
00064      11  WRITE (5,2)
00065      2  FORMAT (1X,34HEigenvector weighting factors are:)
00066      DO 15 IV=1,NS
00067      P(IV)=FLOAT(1)
00068      WRITE (5,4) IV,P(IV)
00069      4  FORMAT (1X,2H(,I2,2H)=,F15.6)
00070      15  CONTINUE
00071      WRITE (5,5)
00072      READ (5,*) KK
00073      IF (KK.LE.0) GO TO 20
00074      READ (5,*) (P(IV),IV=1,NS)
00075      20  WRITE (5,6)
00076      6  FORMAT (1X,14HEnter [dA/dP]:)
00077      READ (5,*) ((DAD(II,IJ),IJ=1,NS),II=1,NS)
00078      WRITE (5,7)
00079      7  FORMAT (1X,14HEnter [dB/dP]:)
00080      READ (5,*) ((DBD(II,IJ),IJ=1,NI),II=1,NS)
00081      CALL SENS
00082      IDG=IDGT
00083      CALL LINV2F (V,NS,10,U,IDG,WKAREA,IER)
00084      CALL UERTST(IER,6HLINV2F)
00085      C  CALL USWFM (10HMATRIX7UT:,10,U,10,NS,NS,4)      ***
00086      CALL tran(U,NS,NS)
00087      C  CALL USWFM (10HMATRIX7U :,10,U,10,NS,NS,4)      ***
00088      CALL SCOST(CJ)
00089      CALL SGRAD
00090      CALL SEARCH(CJ,SCOST,SGRAD,7)
00091      WRITE (IUT*1) ((V(II,IJ),IJ=1,NS),II=1,NS)
00092      WRITE (IUT*2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
00093      WRITE (IUT*4) ((F(II,IJ),IJ=1,NS),II=1,NI)
00094      WRITE (IUT*5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
00095      CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)      ***
00096      WRITE (5,902)
00097      902  FORMAT (1X,44HWISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
00098      READ (5,*) KS
00099      IF (KS.LE.0) GO TO 903
00100      CALL DSPLAY (NS,ZERO)
00101      903  CONTINUE
00102      CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)      ***
00103      CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)      ***
00104      CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)      ***
00105      RETURN
00106      END

```

```

00001 C*****
00002 C*****
00003 SUBROUTINE SCOST(CJ)
00004 C-Function: Calculates the COST function for Sensitivity Reduction.
00005 C-IMSL routines called: (USWFM).
00006 C-Spectral Assignment routines: ZK, and Function routine T .
00007 C-Logical devices; Input Unit: - Output Unit: 5
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL V(10,10),U(10,10),L(10),P(10)
00011 REAL VJ(10),VJ1(10),UJ(10),UJ1(10),LRE(10),LIM(10)
00012 REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
00013 REAL W(10,10),VINV(10,10)
00014 REAL A(10,10),B(10,10),C(10,10)
00015 INTEGER Q
00016 COMMON/SYS/A,B,C,ZERO, IDGT,NS,NI,NO
00017 COMMON/EIG/LRE,LIM/WET/L,P/LEG/U
00018 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00019 CJ1=FLOAT(0)
00020 CJ2=FLOAT(0)
00021 DO 100 J=1,NS
00022 JC=J+1
00023 RELJ=LRE(J)
00024 XIMJ=LIM(J)
00025 c WRITE (5,1) LRE(J),RELJ,LIM(J),XIMJ !!!
00026 c 1 FORMAT (1X,'LRE(J)=RELJ',2F15.6,'LIM(J)=XIMJ',2F15.6) !!!
00027 DO 10 IV=1,NS
00028 VJ1(IV)=FLOAT(0)
00029 UJ1(IV)=FLOAT(0)
00030 VJ(IV)=V(IV,J)
00031 UJ(IV)=U(IV,J)
00032 IF (ABS(XIMJ).LE.ABS(ZERO)) GO TO 10
00033 VJ1(IV)=V(IV,JC)
00034 UJ1(IV)=U(IV,JC)
00035 10 CONTINUE
00036 c CALL USWFV(11HVECTOR VJ:,11, VJ,NS,1,4) !!!
00037 c CALL USWFV(11HVECTOR VJ1:,11,VJ1,NS,1,4) !!!
00038 c CALL USWFV(11HVECTOR UJ:,11, UJ,NS,1,4) !!!
00039 c CALL USWFV(11HVECTOR UJ1:,11,UJ1,NS,1,4) !!!
00040 IF (ABS(XIMJ).LE.ABS(ZERO)) GO TO 20
00041 CJ1=CJ1+L(J)*((T(1,VJ,UJ)-T(1,VJ1,UJ1))**2+((T(1,VJ,UJ)+
00042 1T(1,VJ,UJ1))**2)
00043 GO TO 30
00044 20 CJ1=CJ1+L(J)*T(1,VJ,UJ)**2
00045 30 SUM=FLOAT(0)
00046 DO 50 IQ=1,NS
00047 Q=IQ
00048 NJ=J
00049 CALL ZK(Q,NJ,RELJ,XIMJ,ZRE,ZIM)
00050 SUM=SUM+ZRE**2+ZIM**2
00051 50 CONTINUE
00052 CJ2=CJ2+P(J)*SUM
00053 100 CONTINUE
00054 CJ=CJ1+CJ2
00055 WRITE (5,2) CJ1,CJ2
00056 2 FORMAT (1X,3HJ1=,F15.6,5X,3HJ2=,F15.6)
00057 RETURN
00058 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00001 C*****
00002 C*****
00003 SUBROUTINE SGRAD
00004 C-Function: Calculates the Gradient for Sensitivity Reduction.
00005 C-INSL routines called: UERTST,LINV2F,VMULFF,USWFM.
00006 C-Spectral Assignment routines: PU,DBNORM,PFX,ZK,FRAC,Function T.
00007 C-Logical devices; Input Unit: - Output Unit: 5
00008 C- Storage Unit(s): IU=20+J for J=1,NS.
00009 C-Random Access Files: FORxx.dat where xx=20+J.
00010 C GRADIENT ARRAYAS
00011 REAL G(10,10),U(10,10),PJ1(10,10),PJ2(10,10),L(10),P(10)
00012 REAL VJ(10),VJ1(10),PVJX(10),PVJX1(10),PVJ1X(10),PVJ1X1(10)
00013 REAL UJ(10),UJ1(10),PUKX(10),PUKX1(10),PUK1X(10),PUK1X1(10)
00014 REAL VM(10),VM1(10),UK(10),UK1(10)
00015 REAL PUR(10,10),PUC(10,10),PUC1(10,10)
00016 C NULL SPACE ARRAYS
00017 REAL ML(10,10),NL(10,10)
00018 REAL NLC(10,20),PLC(10,20),MLC(10,20)
00019 REAL STAR(20,20),QL(10,20),RL(10,20)
00020 C AUX. ARRAYS
00021 REAL WKAREA(130)
00022 C MODE 3 ARRAYS
00023 REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
00024 REAL W(10,10),V(10,10),VINV(10,10)
00025 REAL AUX1(10,10),AUX2(10,10),AUX3(10,10),AUX4(10,10)
00026 REAL DAD(10,10),DBD(10,10),DAHD(10,10)
00027 INTEGER Q,FLK,FLJ,FLM
00028 REAL A(10,10),B(10,10),C(10,10)
00029 COMMON/SYS/A,B,C,ZERO, IDGT,NS,NI,NO
00030 COMMON/EIG/LRE,LIM/GR/G/LEG/U/WET/L,P/PJ/PJ1,PJ2
00031 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00032 COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
00033 COMMON/AUX/AUX1,AUX2,AUX3/SEN/DAD,DBD,DAHD/AAUX/AUX4
00034 IGT=IDGT
00035 CALL LINV2F (V,NS,10,VINV,IGT,WKAREA,IER)
00036 CALL UERTST (IER,6HLINV2F)
00037 C CALL USWFM (12HMATRIX VINV:,12,VINV,10,NS,NS,4) !**
00038 IRS=202
00039 J=1
00040 10 FLJ=0
00041 IU=J+20
00042 RELJ=LRE(J)
00043 XIMJ=LIM(J)
00044 IF (ABS(LIM(J)).GT.ABS(ZERO)) FLJ=1
00045 IF (FLJ.EQ.1) GO TO 12
00046 DO 11 IV=1,NS
00047 VJ(IV)=V(IV,J)
00048 UJ(IV)=U(IV,J)
00049 VJ1(IV)=FLOAT(0)
00050 UJ1(IV)=FLOAT(0)
00051 11 CONTINUE
00052 READ (IU'3) ((NL(IJ,IJ),IJ=1,NI),IJ=1,NI)
00053 READ (IU'4) ((ML(IJ,IJ),IJ=1,NI),IJ=1,NI)
00054 C CALL USWFM (10HMATRIX ML:,10,ML,10,NI,NI,4) !**
00055 C CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4) !**
00056 GO TO 14
```

C-2

ORIGINAL PAGE IS  
OF POOR QUALITY

```
00057      12      NI2=2*NI
00058      IS=NS+NI
00059      READ (IU'3) ((NLC(II,IJ),IJ=1,IS),II=1,NS)
00060      READ (IU'4) ((PLC(II,IJ),IJ=1,IS),II=1,NS)
00061      READ (IU'5) ((MLC(II,IJ),IJ=1,IS),II=1,NI)
00062      READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
00063      READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
00064      JC=J+1
00065      DO 13 IV=1,NS
00066      VJ1(IV)=V(IV,JC)
00067      UJ1(IV)=U(IV,JC)
00068      13      CONTINUE
00069      14      DO 110 I=1,NI
00070      KI=I
00071      KJ=J
00072      CALL PFX(KI,KJ,IFLAG)
00073      CALL VMULFF(DBD,AUX3,NS,NI,NS,10,10,AUX4,10,IER)
00074      CALL UERTST (IER,6HVMULFF)
00075      C      CALL USWFM (12HMATRIX AUX4:,12,AUX4,10,NS,NS,4)      !**
00076      IF (FLJ.NE.1) GO TO 115
00077      CALL VMULFF(DBD,W,NS,NI,NS,10,10,AUX4,10,IER)
00078      CALL VMULFF(DBD,AUX3,NS,NI,NS,10,10,W,10,IER)
00079      CALL UERTST (IER,6HVMULFF)
00080      DO 114 II=1,NS
00081      DO 114 IJ=1,NS
00082      AUX3(II,IJ)=W(II,IJ)
00083      114     CONTINUE
00084      C      CALL USWFM (12HMATRIX AUX4:,12,AUX4,10,NS,NS,4)      !**
00085      C      CALL USWFM (12HMATRIX AUX3:,12,AUX3,10,NS,NS,4)      !**
00086      115     IF (FLJ.EQ.1) GO TO 16
00087      DO 15 IV=1,NS
00088      PVJX(IV)=NL(IV,I)
00089      DO 15 IW=1,NS
00090      W(IV,IW)=FLOAT(0)
00091      IF (IW.EQ.J) W(IV,IW)=NL(IV,I)
00092      15      CONTINUE
00093      CALL PU(PUR)
00094      TERM=FLOAT(0)
00095      FLM=0
00096      DO 116 M=1,NS
00097      IF (ABS(LIM(M)).GT.ABS(ZERO)) FLM=1
00098      MC=M+1
00099      DO 216 IV=1,NS
00100      VM(IV)=V(IV,M)
00101      UK(IV)=U(IV,M)
00102      PUKX(IV)=PUR(IV,M)
00103      VM1(IV)=FLOAT(0)
00104      UK1(IV)=FLOAT(0)
00105      PUK1X(IV)=FLOAT(0)
00106      IF (FLM.NE.1) GO TO 216
00107      VM1(IV)=V(IV,MC)
00108      UK1(IV)=U(IV,MC)
00109      PUK1X(IV)=PUR(IV,MC)
00110      216     CONTINUE
00111      IF (FLM.EQ.1) GO TO 316
00112      TERM=TERM+L(M)*(T(2,VM,UK)+T(1,VM,PUKX))*T(1,VM,UK)
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00113          GO TO 116
00114 316  TERM=TERM+L(M)*((T(2,VM,UK)-T(2,VM1,UK1)+T(1,VM,PUKX)-T(1,VM1,PUK1
00115          1X))*T(1,VM,UK)-T(1,VM1,UK1))+T(2,VM,UK1)+T(2,VM1,UK)+T(1,VM,PUK1
00116          2X)+T(1,VM1,PUKX))*T(1,VM1,UK)+T(1,VM,UK1)))
00117 C      WRITE (5,*) TERM          !***
00118 116  CONTINUE
00119          PJI(I,J)=2*(L(J)*T(1,PVJX,UJ)+T(1,VJ,UJ)+TERM)
00120 C      WRITE (5,201) I,J,PJI(I,J)          !***
00121 C 201  FORMAT(1X,'GGGGG I=',I2,'J=',I2,'PJI(I,J)=',F15.6) !***
00122          GO TO 20
00123 16   INI=I+NI
00124          DO 17 IV=1,NS
00125          PVJX(IV)=QL(IV,I)
00126          PVJX1(IV)=QL(IV,INI)
00127          PVJ1X(IV)=RL(IV,I)
00128          PVJ1X1(IV)=RL(IV,INI)
00129          DO 17 IW=1,NS
00130          W(IV,IW)=FLOAT(0)
00131          IF (IW.EQ.J) W(IV,IW)=QL(IV,I)
00132          IF (IW.EQ.JC) W(IV,IW)=RL(IV,I)
00133 17   CONTINUE
00134          CALL PU(PUC)
00135          TERM=FLOAT(0)
00136          FLM=0
00137          DO 117 M=1,NS
00138          IF (FLM.EQ.1) GO TO 317
00139          FLM=0
00140          MC=M+1
00141          IF (ABS(LIM(M)).GT.ABS(ZERO)) FLM=1
00142          DO 217 IV=1,NS
00143          VM(IV)=V(IV,M)
00144          UK(IV)=U(IV,M)
00145          PUKX(IV)=PUC(IV,M)
00146          VM1(IV)=FLOAT(0)
00147          UK1(IV)=FLOAT(0)
00148          PUK1X(IV)=FLOAT(0)
00149          IF (FLM.NE.1) GO TO 217
00150          VM1(IV)=V(IV,MC)
00151          UK1(IV)=U(IV,MC)
00152          PUK1X(IV)=PUC(IV,MC)
00153 217  CONTINUE
00154          IF (FLM.NE.1) TERM=TERM+L(M)*((T(2,VM,UK)+T(1,VM,PUKX))*T(1,VM,UK)
00155          IF (FLM.NE.1) GO TO 117
00156          TERM=TERM+L(M)*((T(2,VM,UK)-T(2,VM1,UK1)+T(1,VM,PUKX)-T(1,VM1
00157          1,PUK1X))*T(1,VM,UK)-T(1,VM1,UK1))+T(2,VM,UK1)+T(2,VM1,UK)
00158          2+T(1,VM,PUK1X)+T(1,VM1,PUKX))*T(1,VM1,UK)+T(1,VM,UK1)))
00159          GO TO 117
00160 317  FLM=0
00161 117  CONTINUE
00162          PJI(I,J)=2*(L(J)*((T(1,PVJX,UJ)-T(1,PVJ1X,UJ1))*T(1,VJ,UJ)-
00163          1T(1,VJ1,UJ1))+T(1,PVJX,UJ1)+T(1,PVJ1X,UJ1))*T(1,VJ1,UJ)+
00164          2T(1,VJ,UJ1))) + TERM)
00165          DO 18 IV=1,NS
00166          DO 18 IW=1,NS
00167          W(IV,IW)=FLOAT(0)
00168          IF (IW.EQ.J) W(IV,IW)=QL(IV,INI)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00169      IF (IW.EQ.JC) W(IV, IW)=RL(IV, INI)
00170      18      CONTINUE
00171      CALL PU(PUC1)
00172      TERM=FLOAT(0)
00173      FLM=0
00174      DO 118 M=1, NS
00175      IF (FLM.EQ.1) GO TO 318
00176      FLM=0
00177      MC=M+1
00178      IF (ABS(LIM(M)).GT.ABS(ZERO)) FLM=1
00179      DO 218 IV=1, NS
00180      VM(IV)=V(IV, M)
00181      UK(IV)=U(IV, M)
00182      PUKX1(IV)=PUC1(IV, M)
00183      VM1(IV)=FLOAT(0)
00184      UK1(IV)=FLOAT(0)
00185      PUK1X1(IV)=FLOAT(0)
00186      IF (FLM.NE.1) GO TO 218
00187      VM1(IV)=V(IV, MC)
00188      UK1(IV)=U(IV, MC)
00189      PUK1X1(IV)=PUC1(IV, MC)
00190      218      CONTINUE
00191      IF (FLM.NE.1) TERM=TERM+L(M)*(T(3, VM, UK)+T(1, VM, PUKX1))*T(1, VM, UK)
00192      IF (FLM.NE.1) GO TO 118
00193      TERM=TERM+L(M)*((T(3, VM, UK)-T(3, VM1, UK1))+T(1, VM, PUKX1)-T(1, VM1,
00194      1, PUK1X1))*T(1, VM, UK)-T(1, VM1, UK1))+T(3, VM, UK1)+T(3, VM1, UK)
00195      2+T(1, VM, PUK1X1)+T(1, VM1, PUKX1))*T(1, VM1, UK)+T(1, VM, UK1))
00196      GO TO 118
00197      318      FLM=0
00198      118      CONTINUE
00199      PJ1(I, JC)=2*(L(JC))*((T(1, PVJX1, UJ)-T(1, PVJ1X1, UJ1))*T(1, VJ, UJ)-
00200      1T(1, VJ1, UJ1))+T(1, PVJX1, UJ1)+T(1, PVJ1X1, UJ1))*T(1, VJ1, UJ1)+
00201      2T(1, VJ, UJ1))+TERM)
00202      PJ2(I, JC)=FLOAT(0)
00203      20      PJ2(I, J)=FLOAT(0)
00204      M=1
00205      40      FLM=0
00206      MC=M+1
00207      IF (ABS(LIM(M)).GT.ABS(ZERO)) FLM=1
00208      DO 45 IV=1, NS
00209      VM(IV)=V(IV, M)
00210      VM1(IV)=FLOAT(0)
00211      IF (FLM.EQ.1) VM1(IV)=V(IV, MC)
00212      45      CONTINUE
00213      TERM1=FLOAT(0)
00214      TIM1=FLOAT(0)
00215      TERM11=FLOAT(0)
00216      TIM11=FLOAT(0)
00217      SUMQ1=FLOAT(0)
00218      SUMQ2=FLOAT(0)
00219      DO 90 Q=1, NS
00220      KC=Q
00221      IF (M.EQ.J) GO TO 50
00222      VQJ=V(Q, J)
00223      VQJ1=FLOAT(0)
00224      IF (FLJ.EQ.1) VQJ1=V(Q, JC)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00225      IF (FLJ.EQ.1) GO TO 48
00226      PQJX=NL(Q,I)
00227      PQJIX=FLOAT(0)
00228      GO TO 49
00229      48  PQJX=QL(Q,I)
00230      PQJIX=RL(Q,I)
00231      PQJX1=QL(Q,INI)
00232      PQJIX1=RL(Q,INI)
00233      49  CALL FRAC(KQ,KJ,M,LRE(M),LIM(M),1,UJ,UJ1,VM,VM1,PQJX,PQJIX,
00234      ITERM1,TIM1)
00235      C   WRITE (5,202) TERM1,FRE,TIM1,FIM,J,I,M,Q      !***
00236      C 202  FORMAT (1X,'TERM1,FRE,TIM1,FIM',4F15.6,/,1X,'J,I,M,Q',4I2) !***
00237      IF (FLJ.NE.1) GO TO 50
00238      CALL FRAC(KQ,KJ,M,LRE(M),LIM(M),1,UJ,UJ1,VM,VM1,PQJX1,PQJIX1,
00239      ITERM11,TIM11)
00240      50  CONTINUE
00241      TERM2=FLOAT(0)
00242      TIM2=FLOAT(0)
00243      TERM21=FLOAT(0)
00244      TIM21=FLOAT(0)
00245      K=1
00246      51  FLK=0
00247      KC=K+1
00248      IF (ABS(LIM(K)).GT.ABS(ZERO)) FLK=1
00249      IF (FLK.NE.1.AND.K.EQ.M) GO TO 58
00250      IF (FLK.EQ.1.AND.K.EQ.M) GO TO 57
00251      DO 52 IV=1,NS
00252      IF (FLJ.NE.1) PUKX(IV)=PUR(IV,K)
00253      IF (FLJ.EQ.1) PUKX(IV)=PUC(IV,K)
00254      PUKIX(IV)=FLOAT(0)
00255      IF (FLK.EQ.1.AND.FLJ.NE.1) PUKIX(IV)=PUR(IV,KC)
00256      IF (FLK.EQ.1.AND.FLJ.EQ.1) PUKIX(IV)=PUC(IV,KC)
00257      PUKX1(IV)=FLOAT(0)
00258      IF (FLJ.EQ.1) PUKX1(IV)=PUC1(IV,K)
00259      PUKIX1(IV)=FLOAT(0)
00260      IF (FLK.EQ.1.AND.FLJ.EQ.1) PUKIX1(IV)=PUC1(IV,KC)
00261      UK(IV)=U(IV,K)
00262      UK1(IV)=FLOAT(0)
00263      IF (FLK.EQ.1) UK1(IV)=U(IV,KC)
00264      52  CONTINUE
00265      VQK=V(Q,K)
00266      VQK1=FLOAT(0)
00267      IF (FLK.EQ.1) VQK1=V(Q,KC)
00268      CALL FRAC(KQ,K,M,LRE(M),LIM(M),1,PUKX,PUKIX,VM,VM1,VQK,VQK1,
00269      IFRE1,FIM1)
00270      CALL FRAC(KQ,K,M,LRE(M),LIM(M),2,UK,UK1,VM,VM1,VQK,VQK1,FRE,FIM)
00271      C   WRITE (5,204) TERM2,FRE,TIM2,FIM,J,I,M,Q,K      !***
00272      C 204  FORMAT (1X,'TERM2,FRE,TIM2,FIM',4F15.6,/,1X,'J,I,M,Q,K',5I2) !***
00273      TERM2=TERM2+FRE+FRE1
00274      TIM2=TIM2+FIM+FIM1
00275      C   WRITE (5,204) TERM2,FRE,TIM2,FIM,J,I,M,Q,K      !***
00276      IF (FLJ.NE.1) GO TO 57
00277      CALL FRAC(KQ,K,M,LRE(M),LIM(M),1,PUKX1,PUKIX1,VM,VM1,VQK,
00278      1VQK1,FRE1,FIM1)
00279      CALL FRAC(KQ,K,M,LRE(M),LIM(M),3,UK,UK1,VM,VM1,VQK,VQK1,FRE,FIM)
00280      TERM21=TERM21+FRE+FRE1

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00281      TIM21=TIM21+FIM+FIP1
00282      57  IF (FLK.EQ.1) K=K+1
00283      58  IF (K.EQ.NS) GO TO 60
00284      K=K+1
00285      GO TO 51
00286      60  CONTINUE
00287      TERM3=FLOAT(0)
00288      TIM3=FLOAT(0)
00289      TERM31=FLOAT(0)
00290      TIM31=FLOAT(0)
00291      IF (M.NE.J) GO TO 70
00292      K=1
00293      61  FLK=0
00294      KC=K+1
00295      IF (ABS(LIM(K)).GT.ABS(ZERO)) FLK=1
00296      IF (FLK.NE.1.AND.K.EQ.J) GO TO 68
00297      IF (FLK.EQ.1.AND.K.EQ.J) GO TO 67
00298      DO 62 IV=1,NS
00299      UK(IV)=U(IV,K)
00300      UK1(IV)=FLOAT(0)
00301      IF (FLK.EQ.1) UK1(IV)=U(IV,KC)
00302      62  CONTINUE
00303      VQK=V(Q,K)
00304      VQK1=FLOAT(0)
00305      IF (FLK.EQ.1) VQK1=V(Q,KC)
00306      CALL FRAC(KQ,K,J,LRE(J),LIM(J),1,UK,UK1,PVJX,PVJ1X,VQK,VQK1,FRE
00307      1,FIM)
00308      C      WRITE (5,205) TERM3,FRE,TIM3,FIM,J,I,M,Q,K      !***
00309      C 205  FORMAT (1X,'TERM3,FRE,TIM3,FIM',4F15.6,/,1X,'J,I,M,Q,K',5I2) !***
00310      TERM3=TERM3+FRE
00311      TIM3=TIM3+FIM
00312      C      WRITE (5,205) TERM3,FRE,TIM3,FIM,J,I,M,Q,K      !***
00313      IF (FLJ.NE.1) GO TO 67
00314      CALL FRAC(KQ,K,J,LRE(J),LIM(J),1,UK,UK1,PVJX1,PVJ1X1,VQK,VQK1,
00315      1FRE,FIM)
00316      TERM31=TERM31+FRE
00317      TIM31=TIM31+FIM
00318      67  IF (FLK.EQ.1) K=K+1
00319      68  IF (K.EQ.NS) GO TO 70
00320      K=K+1
00321      GO TO 61
00322      70  CONTINUE
00323      C      WRITE (5,206) TERM1,TERM2,TERM3,TIM1,TIM2,TIM3      !***
00324      C 206  FORMAT (1X,'TERM1,TERM2,TERM3,TIM1,TIM2,TIM3',/,1X,6F15.6) !***
00325      PZXRE=TERM1+TERM2+TERM3
00326      PZXIM=TIM1+TIM2+TIM3
00327      CALL ZK (KQ,M,LRE(M),LIM(M),ZRE,ZIM)
00328      C      WRITE (5,207) SUMQ1,ZRE,PZXRE,ZIM,PZXIM,J,I,M,Q      !***
00329      C 207  FORMAT (1X,'SUMQ1,ZRE,PZXRE,ZIM,PZXIM',/,1X,5F15.6,'J,I,M,Q',4I2)!*
00330      SUMQ1=SUMQ1+ZRE*PZXRE+ZIM*PZXIM
00331      C      WRITE (5,208) SUMQ1,J,I,M,Q      !***
00332      C 208  FORMAT (1X,'SUMQ1=',F15.6,'J,I,M,Q',4I2)      !***
00333      IF (FLJ.NE.1) GO TO 80
00334      PZXRE=TERM11+TERM21+TERM31
00335      PZXIM=TIM11+TIM21+TIM31
00336      SUMQ2=SUMQ2+ZRE*PZXRE+ZIM*PZXIM

```



ORIGINAL PAGE IS  
OF POOR QUALITY

```
00337      80      CONTINUE
00338      PJ2(I,J)=2*P(M)*SUMQ1+PJ2(I,J)
00339      IF (FLJ.EQ.1) PJ2(I,JC)=2*P(MC)*SUMQ2+PJ2(I,JC)
00340      C        WRITE (5,203) M,I,J,PJ2(I,J)          !***
00341      C 203     FORMAT(1X,'==M=',I2,'==I=',I2,'==J=',I2,'PJ2(I,J)=',F15.6) !***
00342      IF (FLM.EQ.1) M=M+1
00343      IF (M.EQ.NS) GO TO 110
00344      M=M+1
00345      GO TO 40
00346      110     CONTINUE
00347      IF (FLJ.EQ.1) J=JC
00348      IF (J.EQ.NS) GO TO 120
00349      J=J+1
00350      GO TO 10
00351      120     DO 130 II=1,NI
00352      DO 130 IJ=1,NS
00353      G(II,IJ)=PJ1(II,IJ)+PJ2(II,IJ)
00354      130     CONTINUE
00355      C        CALL USWFM (11HMATRIX [G]:,11,G,10,NI,NS,4)      !***
00356      CALL DBNORM (NI,NS)
00357      CALL USWFM (16HGradient matrix:,16,G,10,NI,NS,4)      !***
00358      RETURN
00359      END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003 SUBROUTINE ZK(Q,J,RELJ,XIMJ,ZRE,ZIM)
00004 C-Function: Expression evaluator for Mode 7.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: COMDIV, and Function T.
00007 C-Logical devices; Input Unit: - Output Unit: (5)
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 REAL V(10,10),U(10,10)
00011 REAL VJ(10),VJ1(10),UK(10),UK1(10),LRE(10),LIM(10)
00012 REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
00013 REAL W(10,10),VINV(10,10)
00014 REAL A(10,10),B(10,10),C(10,10)
00015 INTEGER Q,FLJ,FLK
00016 COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00017 COMMON/EIG/LRE,LIM/LEG/U
00018 COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00019 FLJ=0
00020 FLK=0
00021 JC=J+1
00022 ZRE=FLOAT(0)
00023 ZIM=FLOAT(0)
00024 IF (ABS(XIMJ).GT.ABS(ZERO)) FLJ=1
00025 DO 900 K=1,NS
00026 IF (FLK.NE.1) GO TO 10
00027 FLK=0
00028 GO TO 900
00029 10 CONTINUE
00030 C WRITE (5,3) J,K,LRE(K),LIM(K) ***
00031 C 3 FORMAT (1X,'-----J=',I2,'K=',I2,'LAMBDA-K',2F15.6) ***
00032 IF (ABS(LIM(K)).GT.ABS(ZERO)) FLK=1
00033 IF (K.EQ.J) GO TO 900
00034 KC=K+1
00035 DO 100 IV=1,NS
00036 VJ1(IV)=FLOAT(0)
00037 VJ(IV)=V(IV,J)
00038 IF (FLJ.EQ.1) VJ1(IV)=V(IV,JC)
00039 UK(IV)=U(IV,K)
00040 IF (FLK.EQ.1) UK1(IV)=U(IV,KC)
00041 100 CONTINUE
00042 VQK=V(Q,K)
00043 VQK1=FLOAT(0)
00044 C CALL USWFV(11HVECTOR VJ:,11, VJ,NS,1,4) ***
00045 C CALL USWFV(11HVECTOR VJ1:,11,VJ1,NS,1,4) ***
00046 C CALL USWFV(11HVECTOR UK:,11, UK,NS,1,4) ***
00047 C CALL USWFV(11HVECTOR UK1:,11,UK1,NS,1,4) ***
00048 C WRITE (5,2) Q,K,VQK,VQK1
00049 C 2 FORMAT (1X,2HQ=,I2,2HK=,I2,4HVQK=,F15.6,5HVQK1=,F15.6)
00050 IF (FLK.EQ.1) VQK1=V(Q,KC)
00051 IF (FLJ.NE.1.AND.FLK.NE.1) GO TO 200
00052 TRR=T(1,VJ,UK)
00053 TCC=FLOAT(0)
00054 IF (FLK.EQ.1) TCC=T(1,VJ1,UK1)
00055 TRC=T(1,VJ1,UK)
00056 TCR=FLOAT(0)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00057      IF (FLK.EQ.1) TCR=T(1,VJ,UK1)
00058      C      WRITE (5,4) K,J,TRR,TCC,TRC,TCR      !***
00059      C      4      FORMAT (1X,'-----K=',I2,'J=',I2,'TRR---,TCC---,TRC
00060      C      1---,TCR',/,27X,4F15.6)      !***
00061      A1=(TRR-TCC)*VQK-(TRC+TCP)*VQK1
00062      B1=(TRC+TCR)*VQK+(TRR-TCC)*VQK1
00063      A2=RELJ-LRE(K)
00064      B2=LIM(K)-XIMJ
00065      CALL COMDIV(A1,B1,A2,B2,A3,B3)
00066      ZRE=ZRE+A3
00067      ZIM=ZIM+B3
00068      GO TO 900
00069      200      ZRE=ZRE+(T(1,VJ,UK)*VQK)/(RELJ-LRE(K))
00070      900      CONTINUE
00071      C      WRITE (5,1) Q,J,ZRE,ZIM      !***
00072      C      1      FORMAT (1X,1HZ,I2,I2,6H: ZRE=,F15.6,6H: ZIM=,F15.6)      !***
00073      RETURN
00074      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003     SUBROUTINE PU(PUMAT)
00004 C-Function: Returns [PUMAT]=p[U]/p[X] .
00005 C-IMSL routines called: VMULFF,UERTST,(USWFV,USWFM).
00006 C-Spectral Assignment routines: TRAN.
00007 C-Logical devices; Input Unit: -      Output Unit: (5)
00008 C-      Storage Unit: -
00009 C-Random Access Files: -
00010     REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
00011     REAL W(10,10),VINV(10,10),V(10,10),VECTOR(10)
00012     REAL A(10,10),B(10,10),C(10,10)
00013     REAL AUX1(10,10),AUX2(10,10),AUX3(10,10),PUMAT(10,10)
00014     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00015     COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
00016     COMMON/AUX/AUX1,AUX2,AUX3
00017 C     WRITE (5,1) J                !***
00018 C 1   FORMAT (1X,'SUBROUTINE PU, J=',I2)    !***
00019 C     CALL USWFM (10HMATRIX W :,10,W,10,NS,NS,4)    !***
00020     CALL VMULFF(W,VINV,NS,NS,NS,10,10,AUX1,10,IER)
00021     CALL UERTST(10HVMULFF)
00022     CALL VMULFF(VINV,AUX1,NS,NS,NS,10,10,AUX2,10,IER)
00023     CALL UERTST(10HVMULFF)
00024     CALL tran(AUX2,NS,NS)
00025     DO 10 IV=1,NS
00026     DO 10 JV=1,NS
00027     PUMAT(IV,JV)=-AUX2(IV,JV)
00028     10 CONTINUE
00029 C     CALL USWFV(14HVECTOR VECTOR:,14,VECTOR,NS,1,4)    !***
00030     RETURN
00031     END

```

```

00001 C*****
00002 C*****
00003 SUBROUTINE COMDIV(A1,B1,A2,B2,XRE,XIM)
00004 C-Function: COMPLEX DIVISION, XRE+JXIM=(A1+JB1)/(A2+JB2) .
00005 C-IMSL routines called: UERTST,VMULFM,(USWFM,USWFV).
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: - Output Unit: (5)
00008 C- Storage Unit(s): -
00009 C-Random Access Files: -
00010 IF (ABS(B1).GT.FLOAT(0).OR.ABS(B2).GT.FLOAT(0)) GO TO 10
00011 XIM=FLCAT(0)
00012 XRE=A1/A2
00013 GO TO 20
00014 10 XM=SQRT((A1**2+B1**2)/(A2**2+B2**2))
00015 XT=ATAN(B1/A1)-ATAN(B2/A2)
00016 XRE=XM*COS(XT)
00017 XIM=XM*SIN(XT)
00018 20 CONTINUE
00019 C WRITE (5,1) A1,B1,A2,B2,XRE,XIM !!
00020 C 1 FORMAT (1X,F15.6,2H+J,F15.6,1H/,F15.6,2H+J,F15.6,1H=,/ !!
00021 C 1,20X,F15.6,2H+J,F15.6) !!
00022 RETURN
00023 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY.

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003     FUNCTION T(ID,VEC1,VEC2)
00004 C-Function: Evaluates T=<V1,V2> where V1 and V2 are derminrd by
00005 C           the choice of ID,VEC1, and VEC2.
00006 C-IMSL routines called: UERTST,VMULFM,(USWFV,USWFM).
00007 C-Spectral Assignment routines: -
00008 C-Logical devices; Input Unit: -   Output Unit: (5)
00009 C-   Storage Unit(s): -
00010 C-Random Access Files: -
00011     REAL VEC(10),VEC1(10),VEC2(10),DAD(10,10),TX(1,1),DBD(10,10)
00012     REAL A(10,10),B(10,10),C(10,10),DAHD(10,10)
00013     REAL AUX1(10,10),AUX2(10,10),AUX3(10,10),AUX4(10,10)
00014     COMMON/SEN/DAD,DBD,DAHD/AUX/AUX1,AUX2,AUX3/AAUX/AUX4
00015     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00016 C     CALL USWFM (11H[DAHAT/dP]:,11,DAHD,10,NS,NS,4)      !**
00017 C     CALL USWFV (12HVECTOR VEC1:,12,VEC1,NS,1,4)      !**
00018 C     CALL USWFV (12HVECTOR VEC2:,12,VEC2,NS,1,4)      !**
00019     GO TO (1,2,3),ID
00020     1  CALL VMULFM (DAHD,VEC2,NS,NS,1,10,10,VEC,10,IER)
00021     GO TO 10
00022     2  CALL VMULFM(AUX4,VEC2,NS,NS,1,10,10,VEC,10,IER)
00023     GO TO 10
00024     3  CALL VMULFM(AUX3,VEC2,NS,NS,1,10,10,VEC,10,IER)
00025     10 CALL UERTST (IER,6HVMULFM)
00026 C     CALL USWFV (12HVECTOR VEC :,12,VEC ,NS,1,4)      !**
00027     CALL VMULFM (VEC1,VEC,NS,1,1,10,10,TX,1,IER)
00028     CALL UERTST (IER,6HVMULFM)
00029     T=TX(1,1)
00030 C     WRITE (5,11) ID,T      !**
00031 C 11  FORMAT (1X,'ID=',I2,5X,'T=',F15.6)      !**
00032     RETURN
00033     END

```

```

00001 c*****
00002 C*****
00003     SUBROUTINE TRAN(A,IM,IN)
00004 C-Function: Returns the Transpose of [A] in itself.
00005 C-IMSL routines called: -
00006 C-Spectral Assignment routines: -
00007 C-Logical devices; Input Unit: -   Output Unit: -
00008 C-           Storage Unit(s): -
00009 C-Random Access Files: -
00010     REAL A(10,10),AT(10,10)
00011     DO 10 I=1,IM
00012     DO 10 J=1,IN
00013     AT(J,I)=A(I,J)
00014     10 CONTINUE
00015     DO 20 I=1,IN
00016     DO 20 J=1,IM
00017     A(I,J)=AT(I,J)
00018     20 CONTINUE
00019     RETURN
00020     END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 c*****
00002 C*****
00003     SUBROUTINE SENS
00004 C-Function: Calculates d(AHAT)/dP.
00005 C-IMSL routines called: UERTST,VMULFF,(USWFM).
00006 C-Spectral Assignment,routines: -
00007 C-Logical devices; Input Unit: -   Output Unit:   (5)
00008 C-           Storage Unit(s): -
00009 C-Random Access Files: -
00010     REAL DAD(10,10),DBD(10,10),DAHD(10,10),F(10,10),AHAT(10,10)
00011     REAL A(10,10),B(10,10),C(10,10)
00012     COMMON/SEN/DAD,DBD,DAHD/AUG/F,AHAT
00013     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00014     CALL VMULFF (DRD,F,NS,NI,NS,10,10,DAHD,10,IER)
00015     CALL UERTST (IER,6HVMULFF)
00016 C     CALL USWFM (12H[dB/dP]*[F]:,12,DAHD,10,NS,NS,4)   !**
00017     DO 25 II=1,NS
00018     DO 25 IJ=1,NS
00019     DAHD(II,IJ)=DAD(II,IJ)+DAHD(II,IJ)
00020     25 CONTINUE
00021 C     CALL USWFM (11H[dAHAT/dP]:,11,DAHD,10,NS,NS,4)   !**
00022     RETURN
00023     END

```



ORIGINAL PAGE IS  
OF POOR QUALITY

```

00001 C*****
00002 C*****
00003     SURROUTINE FRAC(IQ,IA,IB,RELB,XIMB,ID,UA,UAl,VB,VB1,
00004     1VQA,VQAl,FRE,FIM)
00005 C-Function: Expression evaluator for MODE 7.
00006 C-IMSL routines called: -
00007 C-Spectral Assignment routines: COMDIV, nad Function T.
00008 C-Logical devices; Input Unit: - Output Unit: (5)
00009 C- Storage Unit(s): -
00010 C-Random Access Files: -
00011     REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
00012     REAL W(10,10),VINV(10,10),V(10,10),LRE(10),LIM(10)
00013     REAL A(10,10),B(10,10),C(10,10)
00014     REAL UA(10),UAl(10),VB(10),VB1(10)
00015     COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
00016     COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/EIG/LRE,LIM
00017     FRE=FLOAT(0)
00018     FIM=FLOAT(0)
00019     IA1=IA+1
00020 C     WRITE (5,1) RELB,XIMB,VQA,VQAl,ID     !**
00021 C 1     FORMAT (1X,'RELB,XIMB,VQA,VQAl',4F15.6,'ID=',I2)     !**
00022     IF (IA.EQ.IB) GO TO 99
00023     IF (ABS(XIMB).GT.ABS(ZERO).OR.ABS(LIM(IA)).GT.ABS(ZERO)) GO TO 10
00024     FRE=(VQA*T(ID,VB,UA))/(RELB-LRE(IA))
00025 C     WRITE (5,2) FRE,IQ,IA,IB     !**
00026 C 2     FORMAT (1X,'FRE=',F15.6,5X,'IQ,IA,IB',3I2)     !**
00027     GO TO 99
00028     10     TRR=T(ID,VB,UA)
00029     TCC=FLOAT(0)
00030     IF (ABS(XIMB).GT.ABS(ZERO).AND.ABS(LIM(IA)).GT.ABS(ZERO)) TCC=
00031     1T(ID,VB1,UAl)
00032     TCR=FLOAT(0)
00033     IF (ABS(XIMB).GT.ABS(ZERO)) TCR=T(ID,VB1,UA)
00034     TRC=FLOAT(0)
00035     IF (ABS(LIM(IA)).GT.ABS(ZERO)) TRC=T(ID,VB,UAl)
00036     A1=(TRR-TCC)*VQA-(TCR+TRC)*VQAl
00037     B1=(TCR+TRC)*VQA+(TRR-TCC)*VQAl
00038     A2=RELB-LRE(IA)
00039     B2=LIM(IA)-XIMB
00040     CALL COMDIV (A1,B1,A2,B2,FRE,FIM)
00041     99     RETURN
00042     END

```